

UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**“DISEÑO E IMPLEMENTACION DE UN PROTOTIPO DE MEDICION DE
POTENCIAL ESPONTANEO PARA EL MONITOREO DEL VOLCAN
UBINAS”**

TESIS

PRESENTADA POR:

JUAN ALBERT RAMOS FIGUEROA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2017

Universidad Nacional del Altiplano

FACULTAD DE INGENIERIA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y
SISTEMAS

ESCUELA PROFESIONAL DE INGENIERIA ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MEDICIÓN
DE POTENCIAL ESPONTANEO PARA EL MONITOREO DEL
VOLCÁN UBINAS”

TESIS PRESENTADA POR:

JUAN ALBERT RAMOS FIGUEROA

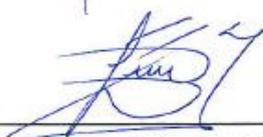
**PARA OPTAR EL TITULO PROFESIONAL DE: INGENIERO
ELECTRÓNICO**

APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE : 
Dr. IVAN DELGADO HUAYTA

PRIMER MIEMBRO : 
M. Sc. DAVID SALINAS MENDOZA

SEGUNDO MIEMBRO : 
Ing. FERDINAND PINEDA ANCCO

DIRECTOR DE TESIS : 
Ing. LUIS ENRIQUE BACA WIESSE

Puno – Perú

2017

Área de investigación: Automatización e instrumentación

Línea de investigación: Instrumentación y control de procesos

DEDICATORIA

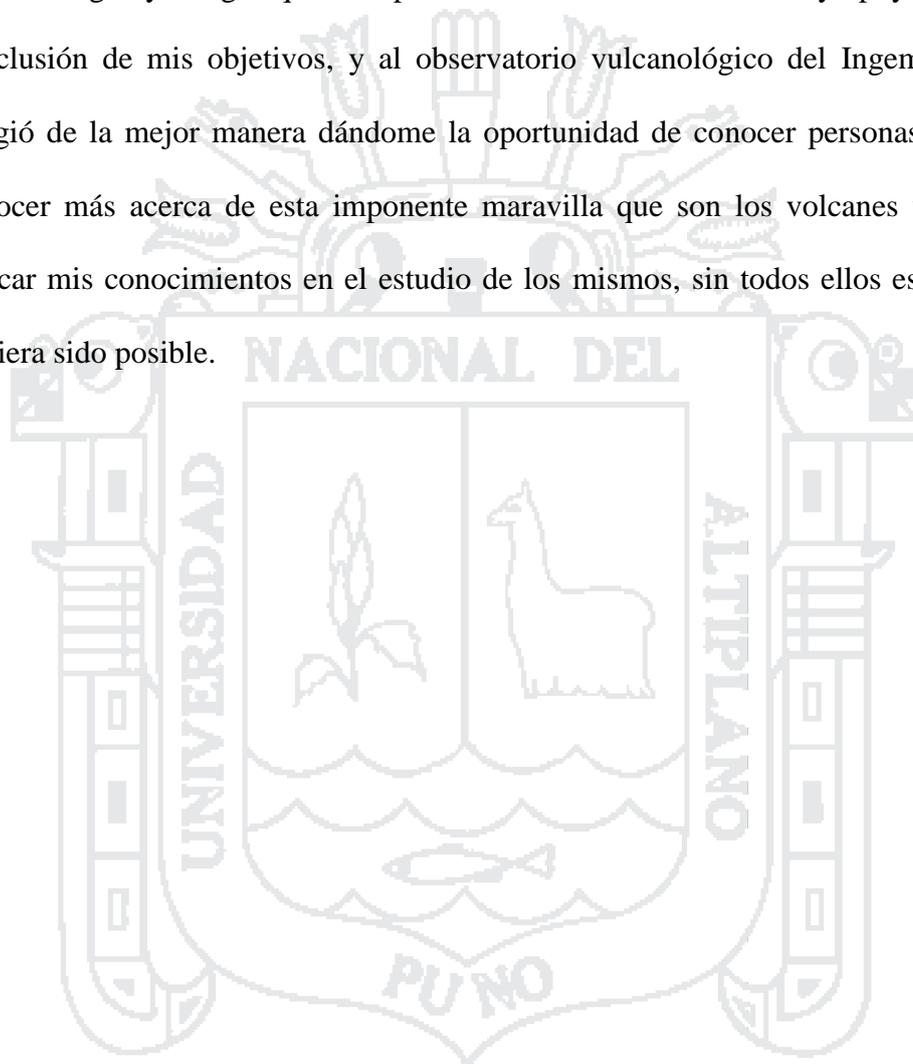
A Dios por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional. A mis padres que aun desde el cielo siguen aportándome inspiración para lograr mis objetivos, a mis hermanos que siempre confiaron en mí y me sacaron adelante dándome ejemplos de superación y entrega.



AGRADECIMIENTOS

Agradezco a Dios por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

A mis amigos y amigas que siempre estuvieron ahí alentándome y apoyándome en la conclusión de mis objetivos, y al observatorio vulcanológico del Ingemmet que me acogió de la mejor manera dándome la oportunidad de conocer personas increíbles y conocer más acerca de esta imponente maravilla que son los volcanes y permitirme aplicar mis conocimientos en el estudio de los mismos, sin todos ellos este trabajo no hubiera sido posible.



INDICE

CAPITULO I.....	20
PLANTEAMIENTO DEL PROBLEMA	20
1.1 PLANTEAMIENTO DEL PROBLEMA	21
1.2 FORMULACION DEL PROBLEMA	21
1.3 JUSTIFICACION DEL PROBLEMA.....	22
1.4 OBJETIVOS DE LA INVESTIGACION.....	23
1.4.1 Objetivo general	23
1.4.2 Objetivos específicos	23
CAPITULO II.....	24
MARCO TEORICO.....	24
2.1 ANTECEDENTES DE LA INVESTIGACION.....	25
2.2 PROSPECCION GEOFISICA.....	27
2.2.1 Métodos de prospección geofísica.....	27
2.2.1.1 Métodos eléctricos	27
2.2.1.2 De campo inducido	27
2.2.1.3 De campo natural	28
2.2.1.4 Fenómenos que perturban la señal de campo.....	29
2.2.1.5 Equipamiento necesario.....	30
2.2.1.5.1 Mili voltímetro.....	33
2.2.1.5.2 Electrodo impolarizables	33
2.3 CONVERTORES ANALOGICO DIGITALES.....	34
2.3.1 Magnitudes analógicas y digitales	34
2.3.2 Codificación binaria.....	36

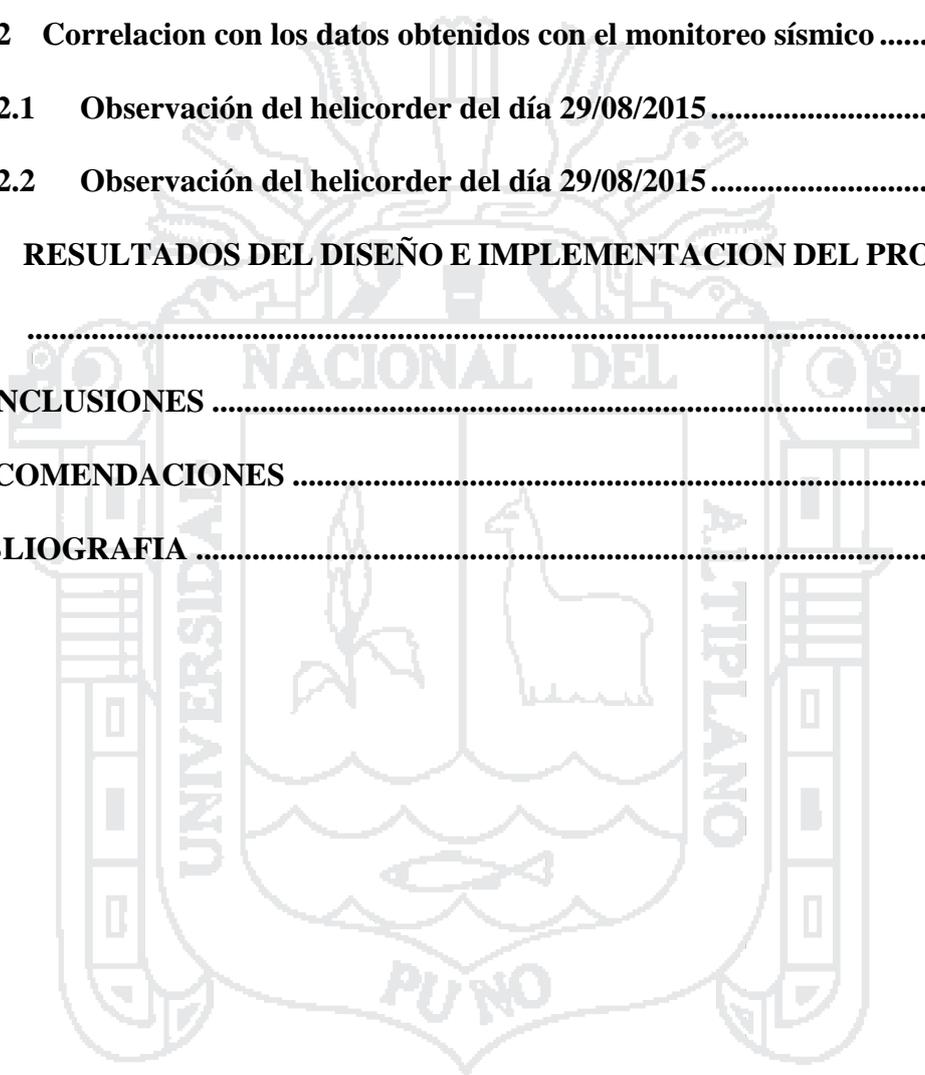
2.3.3	Conversión analógica a digital.....	38
2.3.3.1	Parámetros principales	38
2.3.4	Métodos de conversión analógica digital	39
2.3.4.1	Convertor analógico-digital flash (paralelo).....	39
2.3.4.2	Convertidor analógico-digital de pendiente doble	40
2.3.4.3	Convertor analógico-digital por aproximaciones sucesivas	42
2.3.4.4	Convertor analógico-digital sigma delta	43
2.3.5	Acondicionamiento de señal.....	45
2.3.5.1	Fundamentos del acondicionamiento de señales	46
2.4	TARJETAS DE DESARROLLO	47
2.4.1	ARDUINO.....	48
2.4.1.1	Shields de arduino	49
2.4.2	RASPBERRY PI	50
2.4.2.1	Procesador.....	50
2.4.2.2	Memoria RAM.....	51
2.4.2.3	Conectividad	51
2.4.2.4	Periféricos	51
2.4.2.5	Reloj en tiempo real	51
2.4.2.6	Salidas y entradas de propósito general (GPIO).....	52
2.4.3	SISTEMAS DE COMUNICACIÓN INALAMBRICOS.....	53
2.4.4	SISTEMAS DE COMUNICACIÓN POR SATÉLITE	53
2.4.5	REDES Y SERVICIOS	55
2.4.5.1	REDES.....	55
2.4.5.2	Servicios.....	56
2.4.5.2.1	Servicios de video	57

2.4.5.2.2	Telefonía.....	58
2.4.5.2.3	Servicio de datos.....	58
2.4.6	TELEFONIA CELULAR.....	59
2.4.6.1	REDES GSM	59
2.4.6.2	Redes 3G.....	60
2.4.6.3	Redes 4G.....	61
2.5	FTP (FILE TRANSFER PROTOCOL, PROTOCOLO DE TRANSFERENCIA DE FICHEROS).....	62
2.5.1	SERVIDORES FTP	63
2.5.2	CLIENTES FTP.....	64
2.6	REDES VPN.....	64
2.6.1	VENTAJAS DE UNA VPN PARA UNA ORGANIZACIÓN.....	64
2.7	HIPOTESIS DE LA INVESTIGACION.....	65
2.7.1	Hipótesis general	65
2.7.2	Hipótesis específicas.....	65
2.8	OPERACIONALIZACION DE VARIABLES.....	65
2.8.1	Variable independiente.....	65
2.8.2	Variables dependientes	65
	CAPITULO III.....	66
	MARCO METODOLOGICO	66
3.1	DISEÑO, NIVEL Y TIPO DE LA INVESTIGACION.....	67
3.1.1	Diseño de la investigación.....	67
3.1.2	Nivel de la investigación	67
3.1.3	Tipo de investigación	67
3.2	POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN.....	68

3.2.1	Población.....	68
3.2.2	Muestra.....	68
3.3	UBICACIÓN Y DESCRIPCION DE LA INVESTIGACIÓN.....	68
3.3.1	Ubicación.....	68
3.3.2	Descripción de la investigación.....	70
3.4	TECNICAS E INSTRUMENTOS DE RECOLECCION DE DATOS.....	70
3.5	TECNICAS PARA EL PROCESAMIENTO Y ANALISIS DE DATOS.....	71
3.6	DESCRIPCION DEL PROCESOS DE MEDICIÓN DEL POTENCIAL ESPONTANEO.....	71
3.6.1	Requerimientos básicos para la medición el potencial espontaneo que ha de cumplir el prototipo.....	71
3.7	ELECCION DE TARJETA DE DESARROLLO.....	72
3.8	ELECCION DE LA TECNOLOGIA DE TELECOMUNICACIONES.....	74
3.9	DESCRIPCION DE COMPONENTES.....	75
3.9.1	Determinación del hardware para el prototipo.....	76
3.9.2	Módulo ADS1115.....	77
3.9.2.1	Características.....	79
3.9.2.2	Impedancia de entrada diferencial.....	79
3.9.2.3	Rango de entrada.....	80
3.9.2.4	Programación del módulo ADC ADS1115.....	80
3.9.3	Reloj en tiempo real DS3231.....	82
3.9.3.1	Características.....	83
3.9.3.2	Mapa de registros.....	84
3.9.4	Modem USB HSDPA modelo ZTE MF180.....	84
3.9.4.1	Especificaciones.....	84

3.9.4.2	Tarjeta SIM	85
3.9.5	Plataforma de desarrollo: Raspberry pi 3 B+	86
3.9.6	Memoria micro SD.....	88
3.10	PROCEDIMIENTO DE DISEÑO, E IMPLEMENTACION.....	89
3.10.1	Instalación del sistema operativo.....	89
3.10.1.1	Instalación y configuración de servidor FTP (vsftpd)	97
3.10.2	Conexión de componentes	98
3.10.3	Agregar RTC a Raspberry PI 3 B+.....	99
3.10.4	Realización del código fuente	103
3.10.4.1	Instalación de paquetes de python.....	103
3.10.4.2	Diagrama de flujo del programa.....	106
3.10.4.3	Inicio automático del programa.....	108
3.10.5	Configuración de modem USB GSM/3G	109
3.10.6	Instalación y configuración de VPN.....	116
3.10.6.1	Instalación de hamachi.	117
3.10.7	Realizar script de envío de datos por VPN	119
3.11	CALIBRACION E INSTALACIÓN DEL PROTOTIPO.....	123
3.11.1	Calibración.	123
3.11.1.1	Exactitud del prototipo.	127
3.11.1.2	Precisión del prototipo.	128
3.11.2	Instalación del prototipo.....	128
CAPITULO IV.....		131
ANÁLISIS, E INTERPRETACIÓN DE RESULTADOS DE LA INVESTIGACIÓN.....		131
4.1	ANALISIS E INTERPRETACION DE RESULTADOS.....	132

4.2 RESULTADOS DE LA APLICACIÓN.....	132
4.2.1 Correlacion con los datos obtenidos por el monitoreo visual.....	133
4.2.1.1 Observación del día 29/08/2015.....	133
4.2.1.2 Observación del día 30/08/2015.....	136
4.2.2 Correlacion con los datos obtenidos con el monitoreo sísmico	138
4.2.2.1 Observación del helicorder del día 29/08/2015	138
4.2.2.2 Observación del helicorder del día 29/08/2015	140
4.3 RESULTADOS DEL DISEÑO E IMPLEMENTACION DEL PROTOTIPO..	142
.....	142
CONCLUSIONES	148
RECOMENDACIONES	149
BIBLIOGRAFIA	150



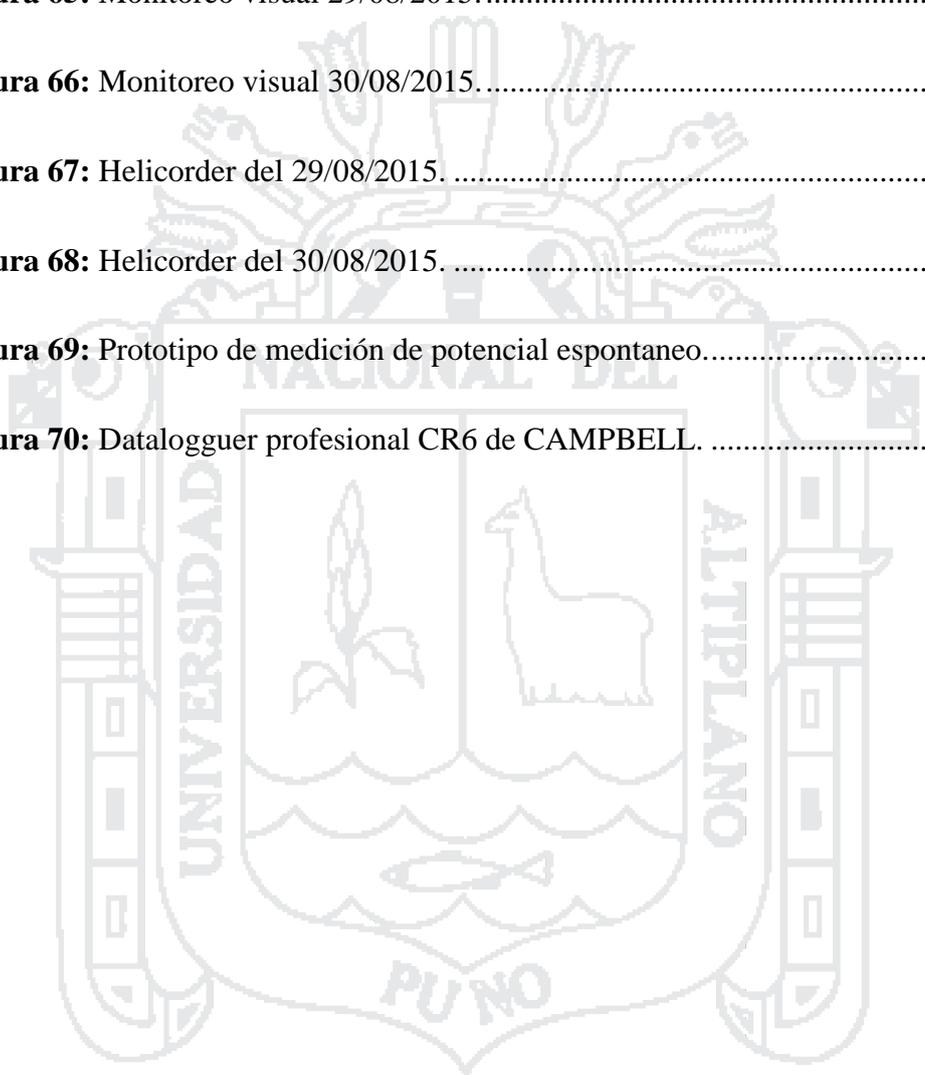
INDICE DE FIGURAS

Figura 1: Prospección eléctrica de campo inducido.	28
Figura 2: Medición de potencial espontaneo.	29
Figura 3: Esquema de medición de potencial espontaneo.	30
Figura 4: Medición del potencial espontaneo.	31
Figura 5: Constitución de un electrodo impolarizable básico.	34
Figura 6: Señal de temperatura continúa.	35
Figura 7: Señales discretas de temperatura.	36
Figura 8: Convertidor ADC flash de 3 bits.	40
Figura 9: Convertidor analógico-digital de rampa doble básico.	41
Figura 10: Conversor A/D por aproximaciones sucesivas.	42
Figura 11: Ilustración de la conversión sigma delta.	44
Figura 12: Diagrama de bloques básico de un convertidor sigma delta.	44
Figura 13: Diagrama de bloques completo de un convertidor sigma delta.	45
Figura 14: Entradas y salidas de propósito general.	52
Figura 15: Satélite cubriendo más de un tercio de la superficie de la tierra.	54
Figura 16: Distribución de frecuencias.	55
Figura 17: Ubicación del observatorio vulcanológico del INGEMMET.	69
Figura 18: Ubicación del punto de medición de potencial espontaneo.	69
Figura 19: Diagrama de bloques del prototipo.	76
Figura 20: Modulo ADC ADS 1115.	78

Figura 21: Diagrama de bloques del ADS1115.....	78
Figura 22: ADS1115.....	81
Figura 23 ADS1115 conectados a un solo maestro.....	82
Figura 24: Modulo Reloj DS3231.....	83
Figura 25: Modem USB ZTE MF180.....	85
Figura 26: Tarjeta SIM de operadora claro.....	86
Figura 27: Raspberry PI 3 B+.....	87
Figura 28: Memoria micro SDHC cat. 10 32 GB.....	88
Figura 29: Ventana de Win32DiskImager.....	91
Figura 30: Grabación finalizada.....	91
Figura 31: Interfaz de Putty.....	92
Figura 32: Escaneando de Raspberry PI 3 B+.....	93
Figura 33: Accediendo a la Raspberry PI.....	94
Figura 34 Consola de Raspberry pi en Putty.....	94
Figura 35 Ventanas de Putty una vez dentro de Raspberry PI.....	95
Figura 36: Archivo de configuración de IP.....	95
Figura 37: Archivo vsftpd.conf.....	98
Figura 38: Esquema de conexión de módulos.....	99
Figura 39: Ventana de configuración de Raspberry.....	100
Figura 40: Ventana de habilitación de del Bus I2C.....	100
Figura 41: Componentes conectados al bus I2C.....	101

Figura 42: RTC instalado en el Kernel.	102
Figura 43: Instalando paquetes de Python.	104
Figura 44: Instalando librerías para ADS1115.	105
Figura 45: Comprobando que el modulo funciona correctamente.	105
Figura 46 Diagrama de flujo para el programa.	107
Figura 47: Modem reconocido correctamente.	110
Figura 48: Modem reconocido como modem USB.	110
Figura 49: Contenido de la carpeta umtskeeper.	111
Figura 50: Interfaz de Sakis3G.	112
Figura 51: Selección de dispositivo.	113
Figura 52: Selección de modem.	113
Figura 53: Selección de APN.	114
Figura 54: Mensaje de confirmación de Sakis3G.	114
Figura 55: Archivo rc.local.	116
Figura 56: Hamachi 2.1.0.139 correctamente instalado.	118
Figura 57: Conexión de Raspberry PI a VPN exitosa.	119
Figura 58: Formato para programar tareas en Cron.	122
Figura 59: Crontab programado para ejecutar upload.sh cada hora.	123
Figura 60: Calibrando el prototipo.	125
Figura 61: Grafica a rango completo.	126
Figura 62: Comparando valores de prototipo y multímetro.	127

Figura 63: Instalación del prototipo en el volcán Ubinas.....	130
Figura 64: Lecturas correspondientes a los días comprendidos entre el 20/08/2015 y el 21/09/2015.	132
Figura 65: Monitoreo visual 29/08/2015.....	135
Figura 66: Monitoreo visual 30/08/2015.....	137
Figura 67: Helicorder del 29/08/2015.	139
Figura 68: Helicorder del 30/08/2015.	141
Figura 69: Prototipo de medición de potencial espontaneo.....	144
Figura 70: Datalogger profesional CR6 de CAMPBELL.	145



INDICE DE TABLAS

Tabla 1: Técnicas e instrumentos de investigación.	70
Tabla 2 Resumen de elección de tarjeta de desarrollo.	73
Tabla 3: Tabla resumen de elección de tecnología de telecomunicación	75
Tabla 4: Impedancia de entrada según el rango de entrada.	79
Tabla 6: Rangos de entrada del PGA.....	80
Tabla 7: Configuración de PGA y su correspondiente rango de entrada.	80
Tabla 8: Conexión del pin ADDR y la correspondiente dirección.	81
Tabla 9: Mapa de registro del DS3231.....	84
Tabla 9: Relación entre voltaje y valores de ADC.	125
Tabla 10: Valores medidos para calcular exactitud y precisión del prototipo.....	127
Tabla 11: Comparación de costo de equipo e instalación.....	146
Tabla 12: Comparación de costo de tecnología de transmisión.	147

INDICE DE ANEXOS

ANEXO 1: CÓDIGO FUENTE EN PYTHON DEL PROTOTIPO.1555

ANEXO 2: CARACTERÍSTICAS DE RASPBERRY PI 3 B+.....1599

ANEXO 3: CARACTERÍSTICAS DEL MÓDULO ADS1115.....16161

ANEXO 4: CARACTERÍSTICAS TÉCNICAS DE CAJAS ID.16868



RESUMEN

El presente trabajo trata del diseño e implementación de un prototipo de medición de potencial espontáneo para el monitoreo del volcán Ubinas, teniendo como objetivo principal la determinación de la utilidad del prototipo en dicha tarea, además de almacenar las lecturas hechas y transmitir las hacia la ciudad de Arequipa haciendo uso de alguna tecnología de transmisión de información inalámbrica, buscando reducir los costos de equipo e instalación al mínimo. Para lograr ello se exploró entre las distintas tarjetas de desarrollo existentes, decantándose en la Raspberry pi 3B+ junto a otros pocos componentes y como tecnología de transmisión de información se eligió usar la red GSM/3G de la operadora Claro, cuya cobertura cubre la zona prevista para la instalación, tras la elección de los componentes y la tarjeta de desarrollo se inició la implementación tratando siempre de ser claro de manera que cualquiera que no tenga conocimientos profundos del tema pueda también implementar el prototipo. La calibración y validación de las mediciones se hicieron usando un multímetro profesional modelo FLUKE 173. Como resultados se presentaron la correlación existente entre las anomalías detectadas por las mediciones de potencial espontáneo y las registradas por los métodos de monitoreo visual y sísmicos, donde se encontraron interesantes resultados sugiriendo que los precursores de un evento volcánico pueden ser detectados con horas de anticipación por el método de potencial espontáneo, lo que ayudaría mucho en la prevención de desastres relacionados al volcán Ubinas. Además se muestra la diferencia de costos de implementación e instalación si se usara un equipo profesional, lo que en costos monetarios y de tiempo es muy significativo.

Palabras claves: Monitoreo, volcanes, potencial espontáneo, Raspberry PI, vigilancia.

ABSTRACT

The present work deals with the design and implementation of a prototype of spontaneous potential measurement for the monitoring of the Ubinas volcano, having as main objective the determination of the utility of the prototype in said task, besides storing the readings made and transmit them to the city of Arequipa making use of some wireless information transmission technology, seeking to reduce equipment and installation costs to a minimum. In order to achieve this, it was explored among the various development cards, choosing the Raspberry pi 3B + along with a few other components and as information technology was chosen to use the GSM / 3G network of the operator Claro, whose coverage covers the area. For the installation, after the choice of components and the development board began the implementation always trying to be clear so that anyone who does not have deep knowledge of the subject can also implement the prototype. Calibration and validation of the measurements were done using a professional multimeter model FLUKE 173. As results, the correlation between the anomalies detected by measurements of spontaneous potential and those recorded by the methods of visual and seismic monitoring were presented, where they were found interesting. Results suggesting that the precursors of a volcanic event can be detected with hours of anticipation by the method of spontaneous potential, which would help a lot in the prevention of disasters related to the Ubinas volcano. It also shows the difference in implementation and installation costs if a professional equipment is used, which in monetary and time costs is very significant.

Keywords: Monitoring, volcanoes, self potential, Raspberry PI, vigilance.

INTRODUCCION

El monitoreo de volcanes es una actividad reciente en nuestro país y sin embargo se han hecho grandes avances en los 4 años de creación del observatorio vulcanológico del Ingemmet (OVI), este avance se debe en gran medida a la colaboración con instituciones extranjeras con mucho más experiencia en el campo, instituciones como la USGS de estados unidos, el IRD de Francia, han sido participes activos en este desarrollo.

La presente tesis busca aportar también a este desarrollo de manera activa, planteándose objetivos específicos y buscando la realización de los mismos, así pues el presente trabajo procura exponer de manera clara el desarrollo y conclusión de un prototipo, partiendo de un planteamiento del problema en el capítulo I donde también se logran las preguntas claves y se establecen los objetivos que dirigirán el curso de la investigación.

El capítulo II trata del estudio exploratorio que se hace de investigaciones previas a esta y el marco teórico que servirán de sustento y punto de partida para la realización de los objetivos planteados en el capítulo 1, además se establecen las hipótesis que se buscara comprobar en los siguientes capítulos.

En el capítulo III establecemos el esquema que deberá seguir la investigación así como su tipo y su alcance, y también se describe el desarrollo del prototipo desde el diseño hasta la instalación en campo.

En el capítulo IV presentamos los resultados del diseño y la instalación, y analizamos los resultados de las lecturas hechas por el prototipo en un periodo de tiempo y las contrastamos con las bases de datos de otras disciplinas para establecer una correlacion entre ellas y de este modo determinar la utilidad del prototipo en el monitoreo del volcán Ubinas.

En el apartado de conclusiones reasentamos los aspectos logrados más relevantes de la investigación, en las recomendaciones nos enfocamos en las mejoras que se le podrían hacer al prototipo así como las acciones que se deberían tomar según algunos casos propuestos, también se habla de las medidas que se deberían tomar en cuenta para mejorar las mediciones y su correlación con otros métodos, y algunas que se deben tener en cuenta en caso de que se pretenda hacer ampliaciones al sistema.

Finalmente se puede encontrar la bibliografía utilizada durante la investigación, siendo que una buena parte se trata de sitios web no será difícil dar con esa bibliografía si se quisiera corroborar lo que se dice en esta investigación y en el apartado final encontramos los anexos que la conforman las hojas de características de los componentes y elementos usados en la implementación del prototipo. Cabe mencionar que la tesis está hecha de tal manera que cualquiera que se interese en el tema, pueda fácilmente implementar el prototipo sin tener mayor experiencia y conocimiento en electrónica y Linux.



CAPITULO I
PLANTEAMIENTO DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Dentro del monitoreo del volcán Ubinas, realizado por el observatorio vulcanológico del INGEMMET (OVI), existen métodos establecidos de monitoreo como el monitoreo sísmico, geoquímico, geodésico. Cuyo objetivo principal es anticiparse a una erupción volcánica con el fin de salvaguardar el bienestar de la población circundante al volcán en cuestión. Pero el trabajo de predicción es bastante complejo y no se puede tener la precisión exacta, por ello se buscan nuevos métodos que permitan mejorar y sirvan de complemento a las anteriormente mencionadas, un efecto con el que se busca experimentar, es el potencial espontaneo que mide el potencial eléctrico del suelo entre dos puntos en el edificio volcánico. Luego es necesario almacenar estos datos y enviarlos hasta una central periódicamente donde serán estudiados. Para lograr este objetivo se necesita contar con equipamiento que es bastante costoso y toma mucho tiempo adquirirlos, pues siendo adquisiciones del estado involucra muchos trámites y papeleos, además siendo este procedimiento experimental, es necesario que el equipamiento sea de bajo coste, fácil implementación y rápida instalación.

1.2 FORMULACION DEL PROBLEMA

Necesitamos diseñar e implementar un prototipo que realice las siguientes tareas: Medir el potencial espontaneo, almacenar datos, y transmitirlos inalámbricamente hacia las instalaciones del observatorio vulcanológico del INGEMMET (OVI) para ello requerimos del hardware y software adecuado.

Debido a la importancia que implica el monitoreo del volcán Ubinas y a la rapidez con que es necesaria su implementación nos preguntamos:

¿En qué medida será de utilidad el prototipo de medición de potencial espontaneo para el monitoreo del volcán Ubinas?

- ¿Qué hardware nos permitirá implementar el prototipo a bajo coste y de manera rápida?
- ¿Qué tecnologías de telecomunicaciones nos permitirá transmitir los datos del prototipo a bajo coste?

1.3 JUSTIFICACION DEL PROBLEMA

Nuestro país se encuentra en lo que se conoce como el cinturón de fuego el cual se caracteriza por concentrar las zonas de subducción más importantes del planeta lo que ocasiona una intensa actividad sísmica y volcánica, además se sabe que en el pasado el Perú se dio la erupción volcánica más grande registrada, la erupción del volcán Huaynaputina, ocurrida en el año 1600, tuvo efectos globales pudiendo ser causa de la hambruna que azotó Rusia entre 1601 y 1603, socialmente hablando los efectos de una erupción volcánica pueden ser terribles habiéndose registrado poblaciones enteras enterradas, incluso pueblos y ciudades abandonadas, pues los materiales que expulsa un volcán puede afectar directamente la salud de la población y los animales, la agricultura. Económicamente hablando una erupción puede afectar grandes proyectos de infraestructura como presas y reservorios de agua, dañando su estructura con los sismos, y contaminando el agua de la que se sirven poblaciones como por ejemplo Arequipa.

Por este motivo en el Perú se vienen monitoreando 4 volcanes, considerados los de mayor peligrosidad por su grado de actividad, influencia económica y poblaciones circundantes las cuales son los volcanes Ubinas y Ticsani ubicadas en la región Moquegua, y Misti y Sabancaya en la región Arequipa. Con el monitoreo continuo de los volcanes mencionados se busca predecir cuándo y cuán grande será una erupción volcánica actualmente con los métodos con los que se cuenta esta tarea aun viene siendo bastante complicada, es por ello que se busca explorar nuevos campos que nos puedan

brindar información que en conjunción con los datos obtenidos por los otros métodos nos ayuden a dar predicciones más exactas, para de esta forma aplicar planes de contingencia que en última instancia beneficiara a la población en general. Técnicamente, encontrar un método que nos brinde información más exacta de las ocurrencias de un volcán sería un paso grande que permitiría investigar más y su aplicación se extendería a todos los volcanes del mundo.

Por otro lado nuestros volcanes, por nuestra geografía, son lugares donde el acceso es muy difícil tomando en muchos casos muchas horas de caminata haciendo el traslado de materiales una tarea difícil, además siempre hay riesgo de robo y si tuviéramos que instalar equipamiento caro implicaría instalaciones más robustas lo que encarecería aún más la ejecución de este procedimiento

Por todo lo mencionado es que se propone el siguiente proyecto de tesis que es el diseñar e implementar un prototipo que nos permita medir el potencial espontaneo, que sea de bajo coste, fácil implementación e instalación.

1.4 OBJETIVOS DE LA INVESTIGACION

1.4.1 Objetivo general

- Determinar si el prototipo de medición de potencial espontaneo a desarrollar será de utilidad para el monitoreo del volcán Ubinas.

1.4.2 Objetivos específicos

- Determinar el hardware mínimo y eficiente que nos permita implementar el prototipo a bajo coste.
- Determinar la tecnología de telecomunicaciones que nos permita transmitir los datos obtenidos por el prototipo de manera periódica a bajo coste.



CAPITULO II
MARCO TEORICO

2.1 ANTECEDENTES DE LA INVESTIGACION

Título: RESISTIVITY AND SELF POTENTIAL CHANGES ASSOCIATED WITH VOLCANIC ACTIVITY: the July 8, 2000 Miyake-jima eruption (Japan)

Autores: J. Zlotnicki, Y. Sasai, P. Yvetot, Y. Nishida, M. Uyeshima, F. Fauquet, H. Utada, Y. Takahashi, G. Donnadieu.

Técnicas empleadas: Prospección por medición de potencial espontáneo, medición de resistividad.

Conclusión principal: La investigación está en torno a la información obtenida de la resistividad y potencial espontáneo para la determinación del comportamiento del sistema hidrotermal del volcán Miyake-jima, que explotó posteriormente, se notó una anomalía negativa grande y por periodo largo de tiempo la interpretación de esta anomalía apunta a que parte del agua del sistema hidrotermal fue drenado debido a un terremoto ocurrido y que habría abierto grietas en el subsuelo, luego se observa una anomalía positiva que indicaría un calentamiento de agua debido al contacto con el magma, todo esto previo a la erupción del volcán Miyake-jima.

Título: METODO DEL POTENCIAL ESPONTANEO APLICADO EN EL VOLCAN UBINAS Y METODOS GEOQUIMICOS APLICADOS EN VIGILANCIA VOLCANICA

Autor: KATHERINE GONZALES ZUÑIGA (INSTITUTO GEOFISICO DEL PERU)

Técnicas empleadas: Prospección por medición de potencial espontáneo.

Conclusión principal: El estudio trata principalmente de establecer y demostrar los lugares donde se encuentran las anomalías de potencial espontáneo, concluyendo que existen anomalías negativas relacionadas a infiltraciones gravitatorias de las aguas

meteóricas hasta la napa acuífera y cuya variación es proporcional a la profundidad del acuífero o la distancia de la zona no saturada, estas anomalías se presentan en las partes bajas del volcán. Por otro lado existen anomalías positivas que se dan en la parte central del edificio volcánico que se correlacionan a zonas de transferencia hídrica y térmica ascendentes, el ascenso de fluidos debido a la aproximación de agua al magma produce una mayor concentración de cationes en las partes altas.

Título: ANALISIS DE SEÑALES SISMICAS REGISTRADAS POR LA RED RADIO TELEMETRICA DEL IGP-AREQUIPA Y POTENCIAL ESPONTANEO APLICADO AL VOLCAN MISTI

Autor: DOMINGO A. RAMOS PALOMINO (INSTITUTO GEOFISICO DEL PERU)

Técnicas empleadas: Prospección por medición de potencial espontaneo y análisis de señales sísmicas.

Conclusión principal: El estudio se basó especialmente en la determinación de anomalías en las diferentes (sísmica, gases y potencial espontaneo) técnicas y la correlación entre ellas, habiéndose encontrado una anomalía de 4000mV en el flanco sur del volcán Misti a una altura de 4200 msnm siendo esta anomalía la más alta registrada en el mundo, interpretándose esta anomalía como el límite del sistema hidrotermal el sector SE del volcán Misti.

Título: TELEMETRIA USANDO REDES DE DATOS DE TELFONIA CELULAR

Autor: J. Marino Dodge.

Conclusión principal: El estudio el estudio se enfocó en la reducción de los costos de monitoreo de parámetros como presión, temperatura, humedad etc. Aplicados a una planta eléctrica, haciendo uso de la red de telefonía y sistemas SCADA.

2.2 PROSPECCION GEOFISICA

Es el arte de aplicar las ciencias físicas al estudio de la parte más superficial de la corteza terrestre, que puede ser explotada por el hombre; en ese sentido, sería la aplicación de los métodos geofísicos de prospección a la búsqueda de toda clase de recursos minerales útiles para el hombre. (Figuerola, 1974)

2.2.1 Métodos de prospección geofísica

A continuación se mencionan los métodos geofísicos existentes de los cuales pondremos especial atención en el método que nos atañe:

- Métodos gravimétricos.
- Métodos magnéticos.
- Métodos sísmicos.
- Métodos eléctricos

De los cuales el que es de nuestro interés es el método eléctrico por tanto:

2.2.1.1 Métodos eléctricos

Son métodos que utilizan las variaciones de las propiedades eléctricas de las rocas y minerales especialmente la resistividad, existen de dos clases principales que son:

2.2.1.2 De campo inducido

Cuando se induce una corriente eléctrica en el terreno para ver su comportamiento ya sea de resistividad o voltaje y corriente.

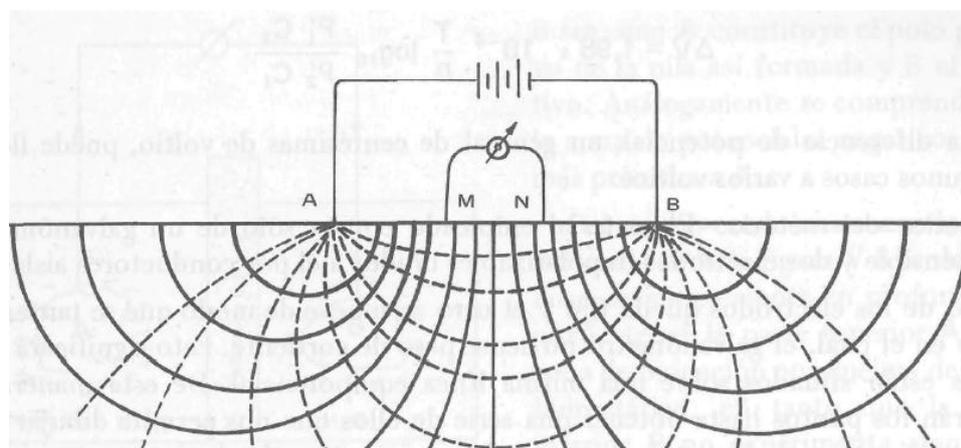


Figura 1: Prospección eléctrica de campo inducido.

Fuente: Figuerola (1974), campo inducido, recuperado de Tratado de geofísica aplicada.

2.2.1.3 De campo natural

Está basado en la observación de los campos eléctricos creados por las fuerzas electromotrices naturales de procedencia electroquímica, filtración de difusión de fluidos como agua y su interacción con las rocas y minerales presentes en el sub suelo. (Lakubovskii & Liajov, 1980). Citando a otro autor:

“La polarización espontanea esta engendrada por las acciones químicas y efectos de contacto entre dos cuerpos o terrenos de distinta naturaleza y resistividad diferente. Dentro de esta modalidad nos fijaremos también en las corrientes que originan diferentes grados de acidez o alcalinidad de las aguas subterráneas.” (Figuerola, 1974, pág. 388)

De las dos ideas anteriores notamos que la composición del terreno es determinante de la creación del potencial espontaneo pero también la composición química del agua subterránea y/o meteórica por filtración o difusión.

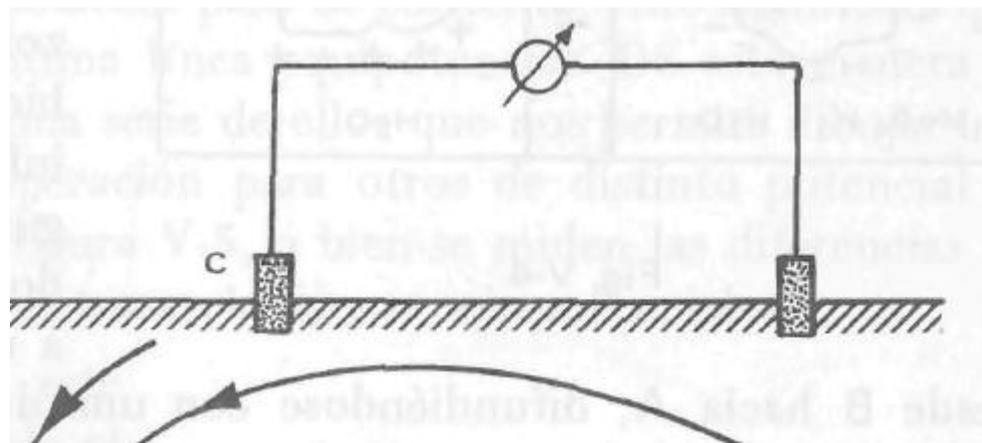


Figura 2: Medición de potencial espontáneo.

Fuente: Figuerola, (1974), campo natural, recuperado de Tratado de geofísica aplicada

2.2.1.4 Fenómenos que perturban la señal de campo

Las medidas de potencial pueden verse afectadas por muchos factores como por ejemplo:

a) **Errores:** Estos pueden ser causados por

- Polarización del electrodo
- Efecto contacto suelo electrodo

Que responden a la variación de parámetros como la temperatura, humedad del suelo, la química del suelo.

b) **Ruido:** Las causas del ruido pueden clasificarse de dos formas, las que generan un potencial bastante estable en el tiempo:

- Efectos topográficos.
- Potenciales electroquímicos.
- Corrosión de elementos metálicos enterrados.
- Actividad biológica.
- Depósitos minerales conductivos.
- Actividad geotérmica.

Y están los que generan un potencial variable en el tiempo:

- Cambios de las propiedades del terreno debido a lluvias, cambios de temperatura o actividades humanas.
- Variaciones del nivel freático.
- Variaciones en los campos de corrosión debido a cambios de las propiedades del suelo.
- Corrientes telúricas.
- Corrientes eléctricas debido a actividades humanas.

2.2.1.5 Equipamiento necesario

Para la realización de las mediciones de potencial son necesarios un milivoltímetro de alta impedancia, dos electrodos de toma a tierra y los cables que conectan los electrodos con el instrumento, se emplean electrodos de construcción especial denominados impolarizables, para estos casos no se usan electrodos metálicos a causa de su gran polarización (Lakubovskii & Liajov, 1980).

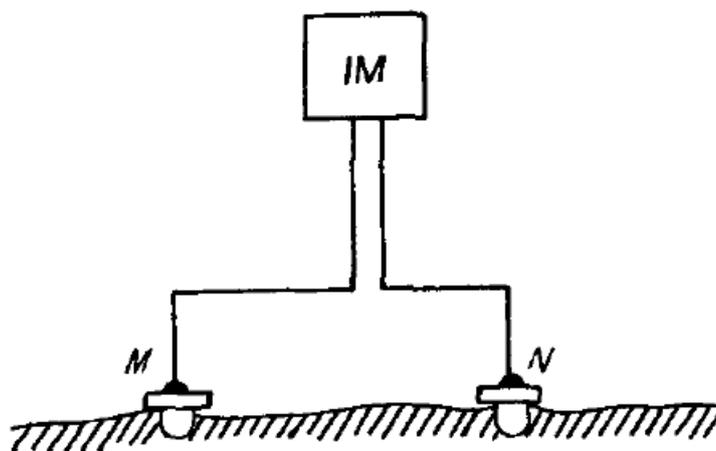


Figura 3: Esquema de medición de potencial espontáneo.

Fuente: Figuerola (1974), medición de potencial espontáneo, recuperado de Tratado de geofísica aplicada

Si la observación del potencial se hiciera con electrodos metálicos, la diferencia de potencial del campo natural se perturbaría ya que la magnitud de polarización de los electrodos metálicos está en el orden del potencial del campo natural y es variable de una medición a otra. La aplicación de un compensador de polarización para anular estas interferencias no es posible puesto que la polarización se superpone a la diferencia de potencial del campo natural que se está midiendo, y no es posible separarlas en el instrumento de medición, con todo esto la eliminación de la polarización debido a los electrodos se realiza por otros medios, el más común es la utilización de electrodos impolarizables. Los cuales están contruidos de manera que el electrodo no tiene contacto directo con el suelo, sino a través de una solución salina del mismo metal del electrodo, así mientras las concentraciones de la solución sea igual, los potenciales en los electrodos serán constantes e iguales, de esta manera la diferencia de los potenciales de los electrodos será cercana a cero y en cuanto a la f.e.m. que aparece debido a difusión de la solución salina hacia el suelo y su mezcla con las sales del suelo es insignificante y no tiene gran influencia en la medición. En la práctica los electrodos impolarizables más usados con los de cobre y plomo, lo expuesto anteriormente se expresa en la siguiente ecuación.

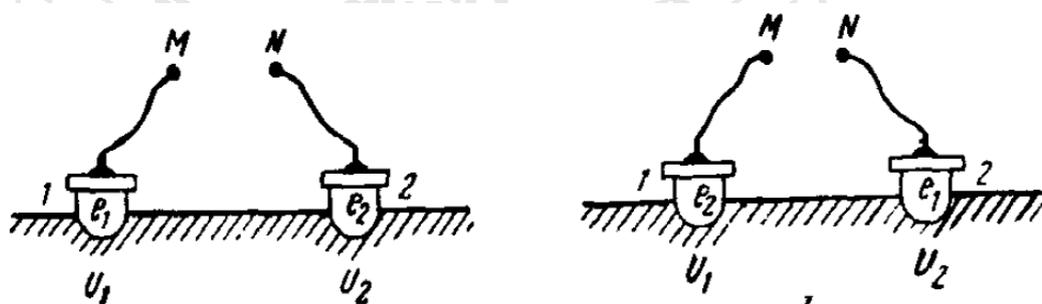


Figura 4: Medición del potencial espontaneo.

Fuente: Figuerola, (1974), potencial espontaneo, recuperdo de Tratado de geofísica aplicada

$$\Delta U_1 = U_M - U_N = (U_1 + e_1) - (U_2 + e_2) = (U_1 - U_2) + (e_1 - e_2) = \Delta U + P$$

Siendo $\Delta U = (U_1 - U_2)$, la diferencia de potenciales desconocida del campo natural y $P = (e_1 - e_2)$, la polarización de los electrodos (diferencia de sus potenciales de electrodos).

Si intercambiamos lugares de los electrodos de la **Figura 4** Y medimos de nuevo la diferencia de potencial entre los mismos puntos tenemos:

$$\Delta U_2 = U'_M - U'_N = (U_1 + e_2) - (U_2 + e_1) = (U_1 - U_2) - (e_1 - e_2) = \Delta U - P$$

De aquí se deduce la diferencia de potencial buscada del campo natural:

$$\Delta U = \frac{(\Delta U_1 + \Delta U_2)}{2} \tag{1}$$

Y la polarización de los electrodos de medición

$$P = \frac{(\Delta U_1 - \Delta U_2)}{2} \tag{2}$$

En caso de igualdad de fuerzas electromotrices de contacto de los electrodos e_1 y e_2 las medidas ΔU_1 y ΔU_2 serán iguales entre si y la polarización $P = 0$.

Si la diferencia de potencial tiene signo positivo, es decir, si $U_1 > U_2$, entonces la corriente circula del 1 al punto 2 y por contrario si la magnitud calculada ΔU es negativa entonces la corriente circula en la dirección del punto 2 al punto 1.

Por tanto a causa de la constancia de la polarización de los electrodos de medición, la misma queda excluida de los resultados de la medición, para el trabajo del

método del potencial natural se debe tener un juego de al menos 5 electrodos impolarizables y estos han de ser preparados desde un día antes por lo menos.

2.2.1.5.1 Mili voltímetro

Debido a que el potencial del suelo son valores pequeños de voltaje que rondan los milivoltios, el instrumento a usar debe contar con ciertas características de impedancia de entrada y resolución, exactitud, precisión, rango, etc. Y debe ser capaz de aceptar valores de voltaje diferenciales.

- La impedancia de entrada debe rondar los 10Mohms de esta manera la impedancia propia del suelo, y del cable no afecta las medidas.
- Debe ser capaz de medir voltajes diferenciales, es decir, debe poder medir voltajes positivos y negativos pues el potencial puede cambiar de signo según se avanza en un perfil o debido a cambios en la química del suelo, o la temperatura del mismo.
- Su rango debe estar entre 0 a +/-3000 voltios, solo en casos excepcionales se ha visto que el potencial ha superado los 4000 mV
- En caso de usar un voltímetro digital la resolución debe ser tal que te permita ver variaciones en el orden de 0.1mV.

2.2.1.5.2 Electrodo impolarizables

Debido a que el contacto directo de electrodos metálicos con el suelo genera un voltaje de polarización enmascarando completamente los voltajes del potencial espontaneo, por tal motivo el contacto de los electrodos con el suelo se hace de forma indirecta, a través de una solución salina del mismo metal haciendo que la concentración en ambos electrodos sea la misma (Gex, 1977), la construcción de los que

usaremos es bastante sencilla, la siguiente figura muestra la constitución de un electrodo no polarizable básico cuyas partes se son:

- Tuvo de PVC 2cm de diámetro interno x 20 cm de largo
- Tapón de goma
- Varilla de cobre 18 cm de largo
- Madera dura y porosa
- Concentración saturada de sulfato de cobre

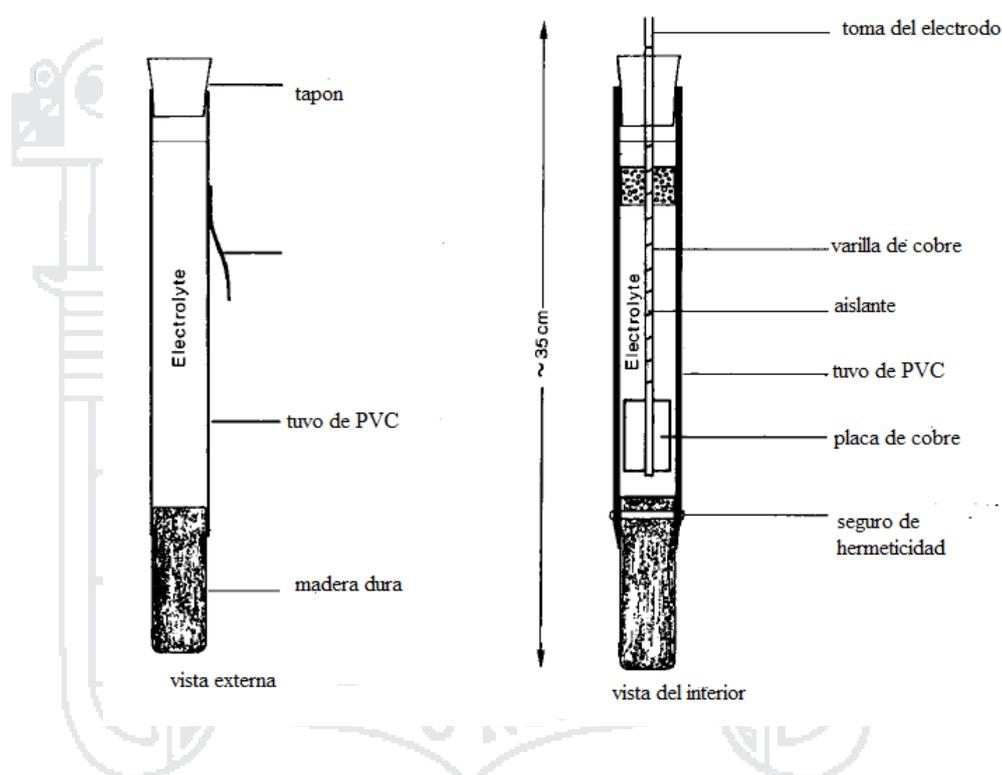


Figura 5: Constitución de un electrodo impolarizable básico.

Fuente: Gex, (1977), recuperado de “la methode de polarisation espontanee”

2.3 CONVERSORES ANALOGICO DIGITALES

2.3.1 Magnitudes analógicas y digitales

Los circuitos electrónicos pueden dividirse en dos grandes grupos, los que trabajan con señales continuas y los que trabajan con señales discretas, una magnitud analógica toma valores continuos mientras que una magnitud analógica toma valores

discretos. La mayoría de las señales que se pueden medir en la naturaleza son continuas así por ejemplo si tomáramos a la temperatura por ejemplo esta no puede cambiar de, digamos 20°C a 30°C instantemente, antes debe pasar por todos los infinitos valores comprendidos en ese intervalo (Floyd, 2006), la **Figura 6** nos muestra una señal analógica continua.

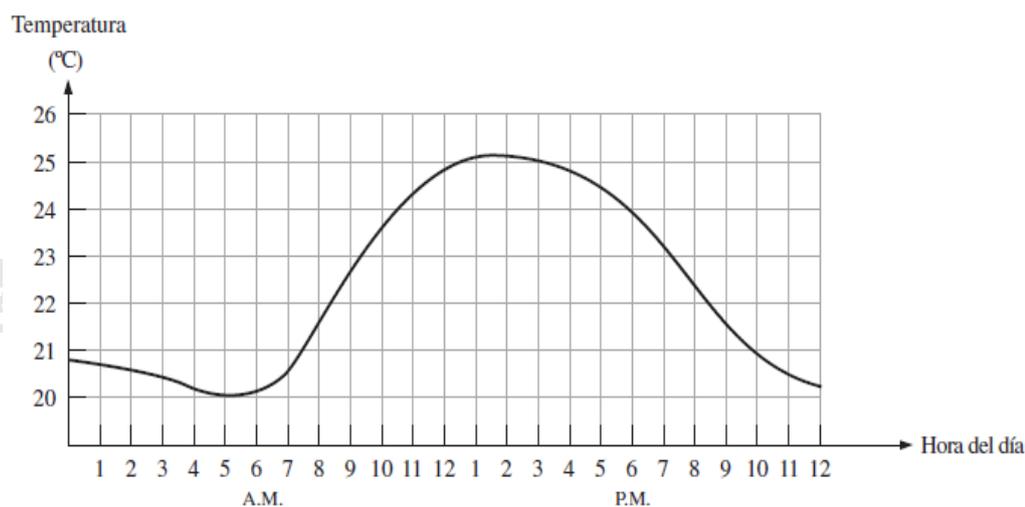


Figura 6: Señal de temperatura continua.

Fuente: Floyd, (2006), señal continua, recuperado de Fundamentos de sistemas digitales.

Si supongamos que medimos esa señal de temperatura y la medimos una vez cada hora, eso quiere decir que tendríamos 24 muestras durante un día como indica **Figura 7**, de esta manera hemos convertido una señal continua en una forma al que se le puede dar un formato digital con valores binarios. (Floyd, 2006)

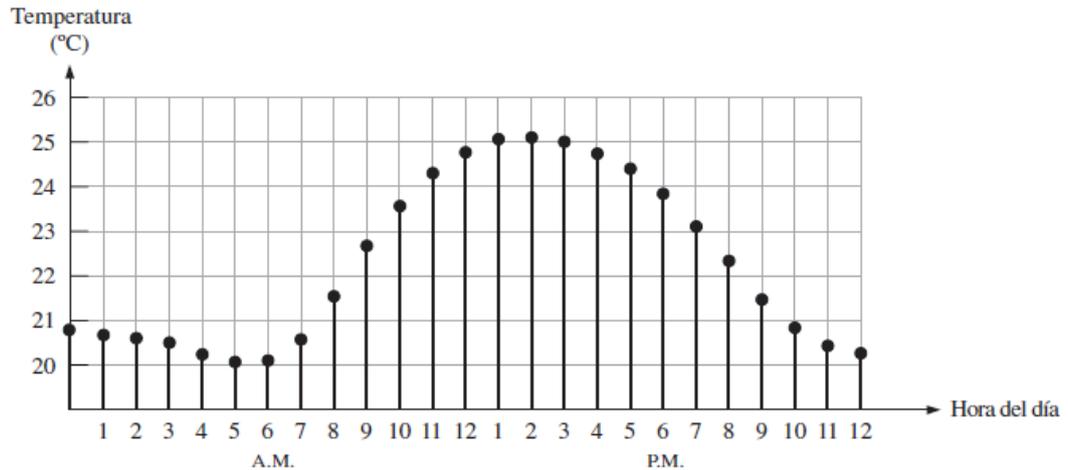


Figura 7: Señales discretas de temperatura.

Fuente: Floyd, (2006), valores cuantizados, recuperado de Fundamentos de sistemas digitales

Los valores binarios tienen la característica de adoptar solo dos valores posibles, “1” o “0”, y dependerá de la tecnología usada el valor de voltaje que corresponda a cada valor posible, así por ejemplo si trabajáramos con sistemas TTL los valores de voltaje que corresponderán a un valor binario “0” están comprendidos entre 0v y 0.8v, y un valor de voltaje comprendido entre 2v y 5v será considerado un “1” binario, y si nos fijamos en el protocolo RS232 un voltaje de -12v representa un valor binario “1” y un voltaje de +12v representa un valor binario “0”.

2.3.2 Codificación binaria

Habitualmente un código binario representa un número que se corresponde con un valor de una variable continua, o una señal de control, siendo esta la que nos interesa también existen otros tipos de codificación que daremos a conocer a continuación:

a) Código binario natural

Se basa en el concepto de numeración posicional con ponderación. Si $d_n \dots d_1$ son valores 0 ó 1, entonces $d_n d_{n-1} \dots d_1 = d_n 2^{n-1} + d_{n-1} 2^{n-2} + \dots + d_1$

Por ejemplo, $10001101 \leftrightarrow 128 + 8 + 4 + 1 = 141$. Los valores d_k se denominan bits (del inglés binary digit). El bit d_n se denomina bit más significativo, y se abrevia MSB (siglas del inglés, most significant bit). El bit d_1 se denomina bit menos significativo, y se abrevia LSB (siglas del inglés, least significant bit).

b) Código binario complemento

Es igual al anterior pero cada bit está invertido. Se utiliza en ciertos casos en que se trabaja con valores lógicos inversos:

$$d_n d_{n-1} \dots d_1 = (1 - d_n) 2^{n-1} + (1 - d_{n-1}) 2^{n-2} + \dots + (1 - d_1)$$

En este ejemplo, $01110010 \leftrightarrow 141$.

c) Códigos con bit de signo

Utiliza el bit más significativo como bit de signo que afecta a los otros bits. Siempre tiene un dígito más que los necesarios para representar el valor absoluto del número. Por ejemplo, en un código de 4 bits:

$$00111 \leftrightarrow +7$$

$$10111 \leftrightarrow -7$$

La desventaja es que no puede manejarse en forma puramente algebraica. Por ejemplo, la suma de dos números negativos debe hacerse con reglas diferentes para el bit más significativo que para el resto. Otra desventaja es que el 0 tiene dos representaciones según que se lo interprete como +0 y -0. La ventaja es la simetría de la representación de números de distinto signo e igual magnitud. Una aplicación habitual son los voltímetros y otros instrumentos digitales, ya que permiten efectuar la decisión sobre la polaridad mediante un comparador, y alimentar directamente el control del segmento del display que representa el signo.

d) Código binario desplazado

Es similar al código binario pero desplazado de modo que el valor 00...0 representa el valor más negativo y 11...1 el más positivo, siendo el 10...0 correspondiente al 0. En general va desde -2^{n-1} a $2^{n-1} - 1$. Por ejemplo, en un código de 4 bits:

$$\begin{aligned} 1111 &\leftrightarrow 7 \\ 1000 &\leftrightarrow 0 \\ 0000 &\leftrightarrow -8 \end{aligned}$$

2.3.3 Conversión analógica a digital

Como se ha visto anteriormente la conversión analógica-digital trata del proceso de convertir una magnitud analógica continua en valores binarios para que estas puedan ser procesadas, almacenadas, etc. Dos parámetros son realmente importantes en un conversor analógico digital, uno es la resolución que es el número de bits y el otro el tasa de transferencia que vendría a ser la velocidad de muestreo (Floyd, 2006) a continuación veremos algunas de las especificaciones que se deben tomar en cuenta al momento de elegir un conversor analógico digital dependiendo siempre de la aplicación en las que se vaya a usar.

2.3.3.1 Parámetros principales

a) Resolución.

Es la cantidad de bits que entrega a su salida luego de completada la conversión. También puede expresarse como el porcentaje o partes por millón (ppm) que representa cada LSB en el rango total de entrada (Miyara, S.F.). Por ejemplo, un conversor de 12 bits tiene una resolución de $\left(\frac{1}{2^{12}-1}\right) * 100 = 0.0244\%$ o dicho de otra forma 244 ppm.

Una manera fácil de identificar la resolución de un ADC, es el número de bits con el que hace la conversión.

b) Velocidad de muestreo

Determina la máxima frecuencia de muestreo que puede realizar, la que nos indica la frecuencia máxima de la señal que podemos digitalizar, el teorema de Nyquist-Shannon establece que la un señal analógica puede ser correctamente reconstruida si la frecuencia de muestreo es como mínimo el doble de la frecuencia de la señal a muestrear.

$$f_s \geq 2f_m \quad (3)$$

Donde f_s es la frecuencia de muestreo y f_m es la frecuencia de la señal muestreada.

2.3.4 Métodos de conversión analógica digital

2.3.4.1 Conversor analógico-digital flash (paralelo)

El método flash utiliza comparadores que comparan una serie de tensiones de referencia con la tensión de entrada analógica. Cuando la tensión analógica sobrepasa a la tensión de referencia de un comparador determinado, se genera un nivel ALTO. La **Figura 8** presenta un convertidor de 3 bits que utiliza siete circuitos comparadores; no se necesita comparador para el caso de que todas las comparaciones sean cero. En general, se requieren $2^n - 1$ comparadores para la conversión a un código binario de n bits. El número de bits empleado en un ADC es su resolución. Una de las desventajas del ADC flash es el gran número de comparadores necesarios para un número binario de tamaño razonable. Su principal ventaja es que tiene un tiempo de conversión rápido, gracias a su alta tasa de transferencia, la cual se mide en muestras por segundo. La tensión de referencia de cada comparador se establece mediante un circuito divisor de

tensión resistivo. La salida de cada comparador se conecta a una entrada del codificador de prioridad. El codificador se habilita mediante un impulso aplicado a la entrada de habilitación EN, y el código de tres bits que representa el valor de la entrada analógica se presenta en las salidas del codificador. El código binario queda determinado por la entrada de mayor orden que se encuentre a nivel ALTO. La frecuencia de los impulsos de habilitación y el número de bits del código binario determinan la precisión con la que la secuencia de códigos digitales representa la entrada del ADC. Debe haber un pulso de habilitación por cada nivel de muestreo de la señal de entrada.

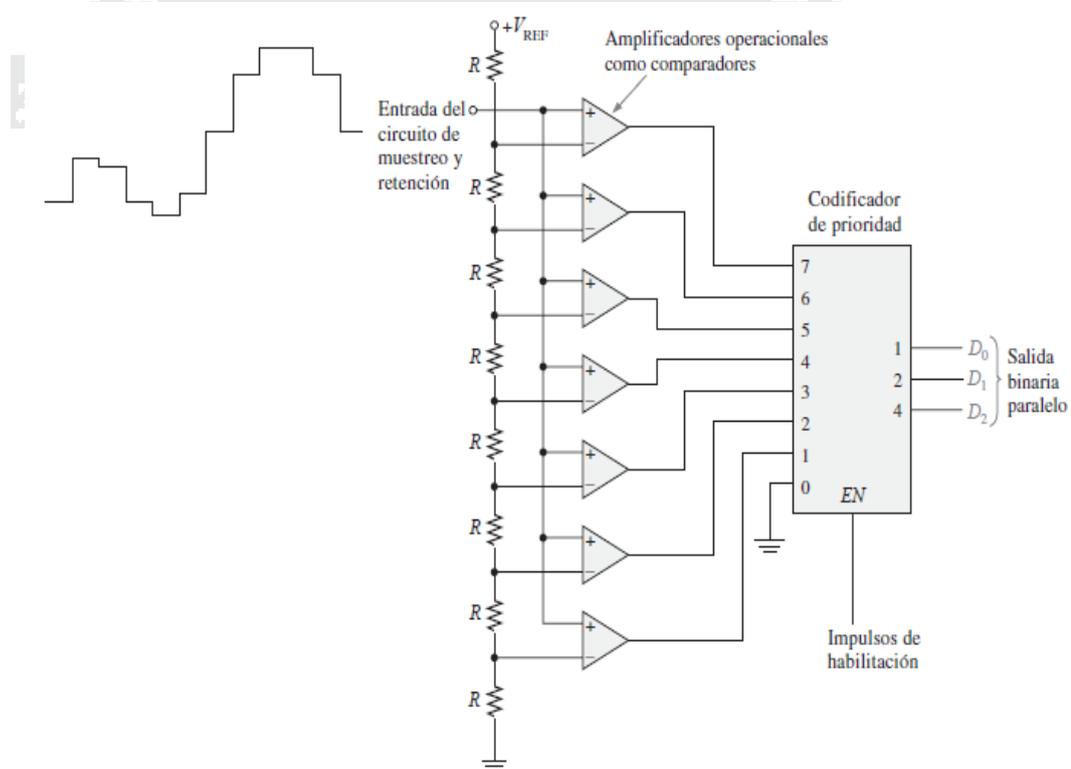


Figura 8: Convertidor ADC flash de 3 bits.

Fuente: Floyd, (2006), Convertidor ADC flash de 3 bits recuperado Fundamentos de sistemas digitales.

2.3.4.2 Convertidor analógico-digital de pendiente doble

El ADC de pendiente doble se utiliza comúnmente en voltímetros digitales y otros tipos de instrumentos de medida. Utilizan un generador de rampa (integrador) para

generar la característica de pendiente doble. En la **Figura 9** se presenta un diagrama de bloques de un ADC de pendiente doble.

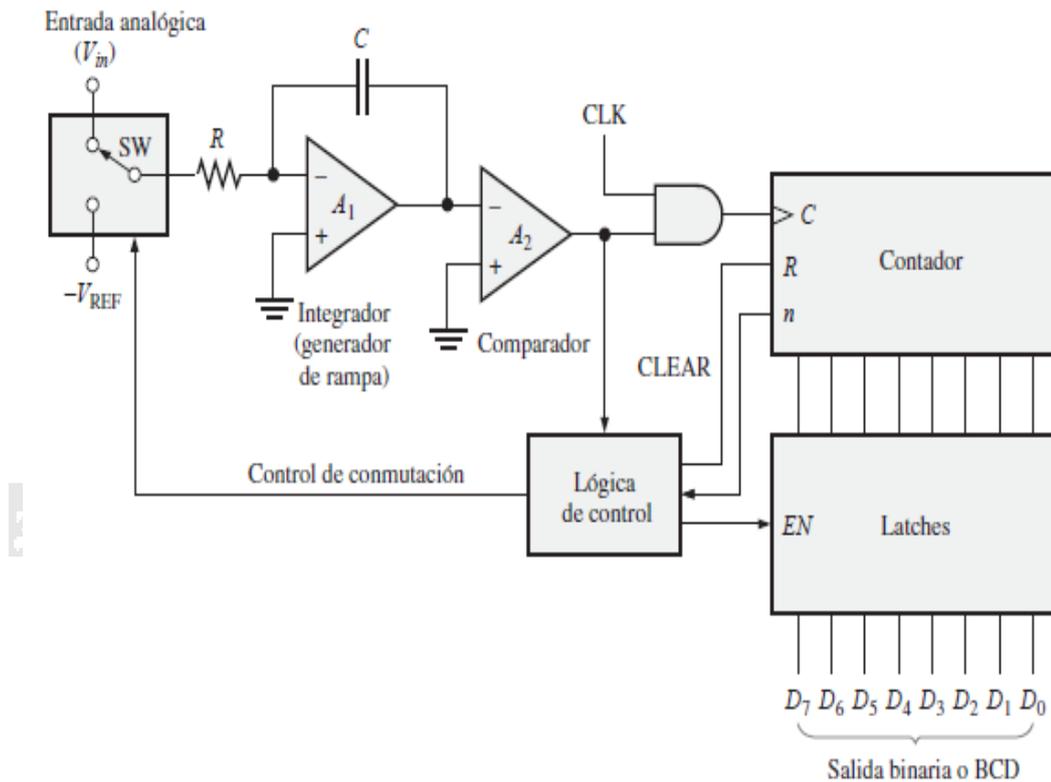


Figura 9: Convertidor analógico-digital de rampa doble básico.

Fuente: Floyd,(2006), convertidor analógico digital derampa doble basico.

Su funcionamiento es relativamente sencillo, si consideramos las condiciones iniciales en los operaciones con valores 0 y activamos el SW haciendo que la entrada de A_1 se conecte a V_{in} el condensador empezará a cargarse de forma lineal hasta un voltaje proporcional a V_{in} , en este instante el switch SW conecta la entrada de A_1 a $-V_{ref}$, y el condensador describe una rampa positiva que va desde el valor $-V$, de la salida del operacional A_1 , hasta un valor 0 donde el contador se detiene y su información se almacena en los latches, este valor de cuenta es proporcional al voltaje de entrada V_{in} , una de las principales desventajas de este método es que dependiendo del valor V_{in} de entrada el tiempo que toma cada muestra puede variar.

2.3.4.3 Conversor analógico-digital por aproximaciones sucesivas

Quizás el método de conversión A/D más ampliamente utilizado es el de las aproximaciones sucesivas. Tiene un tiempo de conversión mucho menor que la conversión de pendiente doble, aunque es más lento que el método flash. Asimismo, el tiempo de conversión es fijo para cualquier valor de la entrada analógica. La muestra un diagrama de bloques básico de un ADC por aproximaciones sucesivas de 4 bits. Está formado por un DAC, un registro de aproximaciones sucesivas (SAR, Successive-Approximation Register) y un comparador. Su funcionamiento básico es el siguiente: los bits de entrada al DAC se habilitan (se ponen a 1) de uno en uno sucesivamente, comenzando por el bit más significativo (MSB). Cada vez que se habilita un bit, el comparador produce una salida que indica si la tensión analógica de entrada es mayor o menor que la salida del DAC. Si la salida del DAC es mayor que la señal de entrada, la salida del comparador está a nivel BAJO, haciendo que el bit en el registro pase a cero. Si la salida es menor que la entrada, el bit 1 se mantiene en el registro. El sistema realiza esta operación con el MSB primero, luego con el siguiente bit más significativo, después con el siguiente, y así sucesivamente. Después de que todos los bits del DAC hayan sido aplicados, el ciclo de conversión estará completo.

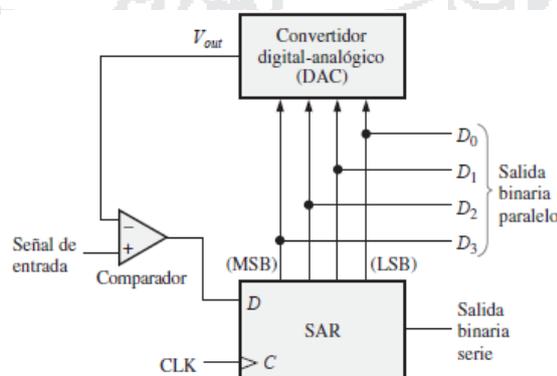


Figura 10: Conversor A/D por aproximaciones sucesivas.

Fuente: Floyd, 2006, Conversor A/D por aproximaciones sucesivas. Recuperado de Fundamentos de sistemas digitales.

2.3.4.4 Conversor analógico-digital sigma delta

Sigma-delta es un método muy extendido de conversión analógico-digital, particularmente cuando se utilizan señales de audio en el campo de las telecomunicaciones. El método está basado en la modulación delta, en la que se cuantifica la diferencia entre dos muestras sucesivas (es decir, se cuantifica el incremento o decremento sufrido por la señal); otros métodos utilizados en los convertidores ADC están basados en el valor absoluto de cada muestra. La modulación delta es un método de cuantificación de un bit. La salida de un modulador delta es un flujo de datos de un único bit en el que el número relativo de 1s y 0s indica el nivel o amplitud de la señal de entrada. El número de 1s a lo largo de un cierto número de ciclos de reloj establece la amplitud de la señal durante dicho intervalo. Un número máximo de 1s corresponde a la tensión de entrada positiva más alta. Un número de 1s igual a la mitad del máximo se corresponde con una tensión de entrada igual a cero. Si no hay ningún 1 (si todos son 0s), lo que tenemos es una tensión de entrada negativa de máxima amplitud. Esto se ilustra de manera simplificada en la **Figura 11**. Por ejemplo, suponga que hay 4096 1s durante el intervalo en el que la señal de entrada presenta un máximo positivo. Puesto que el cero es el punto medio del rango dinámico de la señal de entrada, aparecerán 2048 1s durante el intervalo en que esa señal es cero. Cuando la señal de entrada presenta un máximo negativo no habrá ningún 1 durante el intervalo. Para los niveles de señal intermedios, el número de 1s es proporcional al nivel de la señal.

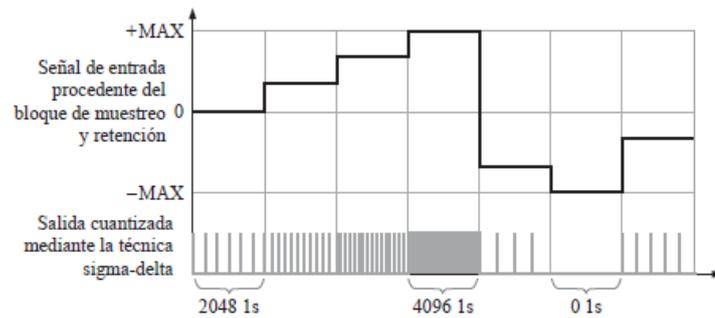


Figura 11: Ilustración de la conversión sigma delta.

Fuente: Floyd, 2006 recuperado de Fundamentos de sistemas digitales

- **Diagrama de bloques de un convertor sigma delta**

El diagrama de bloques básico de la **Figura 12** permite implementar el proceso de conversión ilustrado en la **Figura 11**. La señal analógica de entrada y la señal analógica correspondiente al flujo de bits cuantizado resultante de la conversión en el DAC, dentro del bucle de realimentación, se aplican al punto de suma (Σ). La señal diferencia (Δ) saliente de Σ se integra y el ADC de 1 bit incrementa o decremento el número de 1s dependiendo de la señal diferencia. Esta acción trata de que la señal cuantizada realimentada se asemeje lo más posible a la señal analógica entrante. El cuantizador de 1 bit es esencialmente un comparador, seguido de un latch.

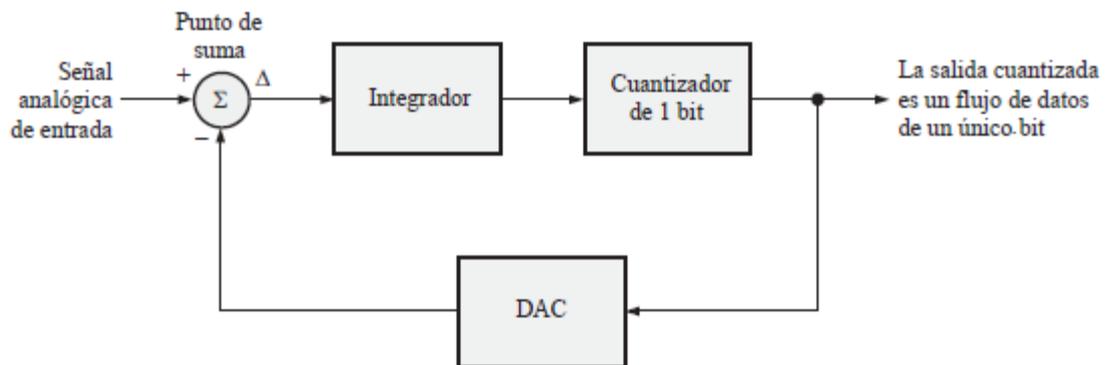


Figura 12: Diagrama de bloques básico de un convertidor sigma delta.

Fuente: Floyd, (2006), diagrama de bloques basico de un convertidor sigma delta, recuperado de Fundamentos de sistemas digitales.

Para completar el proceso de conversión sigma-delta utilizando una técnica concreta, el flujo de datos de un único bit se convierte en una serie de códigos binarios, como se muestra en la **Figura 13** El contador cuenta los 1s en el flujo de datos cuantizado durante cada uno de los sucesivos intervalos. El código almacenado en el contador representa entonces la amplitud de la señal analógica de entrada para cada intervalo. Estos códigos se enclavan en el latch para su almacenamiento temporal. La salida del latch es una serie de códigos de n bits, que representan de manera completa la señal analógica. (Floyd, 2006)

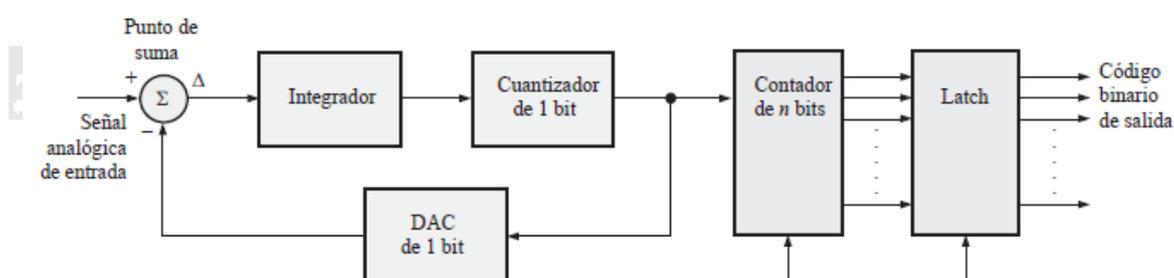


Figura 13: Diagrama de bloques completo de un convertidor sigma delta.

Fuente: Floyd, (2006), recuperado de Fundamentos de sistemas digitales.

2.3.5 Acondicionamiento de señal

Las necesidades de acondicionamiento de las señales varían ampliamente dependiendo de la funcionalidad del sensor, por lo que ningún instrumento puede proporcionar todo tipo de acondicionamiento para todos los sensores. Por ejemplo, los termopares producen señales de muy baja tensión, lo cual requiere liberalización, amplificación y filtrado, mientras que las galgas extensiométricas y acelerómetros necesitan excitación. Otras señales pueden no necesitar ninguna de ellas, sino que se basan en gran medida en el aislamiento de alta tensión. La clave para lograr un exitoso sistema de acondicionamiento de señales es entender los circuitos que son necesarios

para garantizar una medida precisa cualquiera que sea la combinación de canales.
(National instruments, S.F.)

2.3.5.1 Fundamentos del acondicionamiento de señales

La mayoría de las señales necesitan ser acondicionadas antes de que puedan ser digitalizadas, así tenemos sensores que generan voltajes muy pequeños como los termopares, y también están lo que requieren tener alimentación como los RTD, acelerómetros, galgas extensiométricas, etc. A continuación se presentan los principales tipos de acondicionamiento que se hace a una señal antes de que puedan ser digitalizadas.

- a) **Amplificación:** Los amplificadores incrementan la magnitud de la señal al rango de entrada de los convertidores analógico- digital (ADC), mejorando así la resolución y la sensibilidad, además al estar los amplificadores cerca de la fuente de señal mejoran la relación señal-ruido, incrementando la magnitud del voltaje antes de que se vea afectado por el ruido ambiental o de otro tipo, los sensores típicos que necesitan ser amplificados con los termopares o los de deformación como las galgas extensiométricas.
- b) **Atenuación:** Es un proceso contrario a la amplificación, reduce la señal a digitalizar al rango de voltaje que admite el ADC. Generalmente se atenúan señales de voltaje cuyo valor es mayor a 5v.
- c) **Filtrado:** Los filtros rechazan ruido de una determinada frecuencia, generalmente se usan cuando se miden señales cambiantes en el tiempo que es susceptible a ruido de alta frecuencia por ejemplo los 60Hz de la red eléctrica, para el cual se usa un filtro paso bajos, otro de los usos de los filtros es el de evitar el aliasing de señales de alta frecuencia, los filtros anti-aliasing

son un tipo de filtros paso bajo de banda plana y caída rápida, este tipo de filtros son ideales para señales de sonido y vibración.

- d) **Aislamiento:** Cuando las señales se pasan excesivamente del rango del ADC es necesario aislarlas de alguna manera para que no dañen al sistema o los operadores, por este motivo son necesarios el aislamiento y la atenuación para evitar efectos de sobre picos. También se usan en sensores que no comparten una tierra común.
- e) **Excitación:** Algunos sensores como galgas extensiométricas, sensores térmicos RTD, termistores, requieren una excitación externa ya sea de voltaje o corriente, por ejemplo los termistores y RTD requieren una corriente para poder generar un voltaje medible con el ADC, y los acelerómetros que muchas veces cuentan con un amplificador que necesita ser excitado.
- f) **Linealización:** Muchos sensores no están linealmente relacionados con la magnitud física que miden, por ejemplo los termopares. Por este motivo es necesario crear una manera de linealizar la respuesta del sensor, esto generalmente se hace con un software pues hacerlo en hardware es bastante complicado.
- g) **Compensación:** Hay algunos casos donde la influencia de los cables y conexiones alteran el valor real de la magnitud que se desea medir, por ejemplo los termopares que dependen mucho del material del cable, materiales de los puntos de conexión, porque se generan voltajes que pueden afectar las magnitudes de la señal que se desea medir.

2.4 TARJETAS DE DESARROLLO

Desde hace algunos años se han venido desarrollando una innumerable cantidad de tarjetas de desarrollo tras el éxito de Arduino, muchas de ellas basadas en

microcontroladores AVR de ATMEL, y algunas en el ya casi olvidado PIC de Microchip, tras ellas aparecieron las computadoras de placa reducida o SBC siendo la primera de estas la Raspberry PI que está basado en procesadores ARM mucho más potentes y capaces de correr alguna distribución de Linux, y al igual que sucedió con Arduino aparecieron muchas otras tarjetas que pretendían competir con la Raspberry, agregándoles lagunas características que esta no cumpliera, toda esta variedad abre un abanico de opciones que hacen más difícil la elección para un proyecto determinado, pero para que una tarjeta de desarrollo llegue a tener éxito no solo es necesario dotarle de un mejor procesador o microcontrolador u otras características que otra tarjeta no tiene, también es muy importante la comunidad de desarrolladores que tiene detrás, es por este motivo que muchas tarjetas que tienen muy buenas prestaciones han desaparecido o están condenadas a desaparecer.

2.4.1 ARDUINO

Arduino es una plataforma de electrónica que se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso.

La primera placa Arduino fue introducida en 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales. Buscaba desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que usan microcontroladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura y por lo que tampoco lo es su set de

instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.

2.4.1.1 Shields de arduino

Así mismo arduino cuenta con shields, que son placas impresas que se conectan en la parte de arriba del arduino que incrementan sus capacidades permitiendo que arduino pueda leer y escribir memorias SD, controlar motores y sensores, controlar señales de alto voltaje, contar con un reloj en tiempo real, conectarse a internet etc. Siendo muy extensa la lista de estos shields mencionaremos solo los más comunes y fáciles de encontrar.

- **Shield ethernet:** Permite a arduino conectarse a una red aunque debido a la poca memoria de programa de arduino es poco lo que se puede hacer.
- **Shield de relays:** Arduino es capaz de encender y apagar reles que pueden manejar cargas de potencia considerable, como motores, focos, etc.
- **Datalogger Shield:** Este Shield contiene un lector de memorias SD y un reloj en tiempo real (RTC), el cual le da la capacidad de almacenar una gran cantidad de datos que pueden ser obtenidos por uno o varios sensores y también le permite conservar la hora a pesar de ser desconectado.
- **Lcd Shield:** Este Shield contiene un LCD y un interfaz I2C por lo que no usa muchos pines del arduino además cuenta con 4 pulsadores que pueden ser programados para ingresar información manualmente.
- **GSM Shield:** Este Shield le da a arduino la capacidad de conectarse a una red celular GSM permitiéndole enviar y recibir mensajes SMS incluso hacer llamadas, este Shield incluye un GPS que permite saber la ubicación y hora de arduino en todo momento, mientras haya señal de GPS.

- **Motor control Shield:** Este Shield incluye dos driver L293 capaces de controlar hasta 4 motores independientemente, y un ULN2003 que es un array de transistores con él puede manejar cargas de elevado amperaje y un voltaje mayor al voltaje del propio arduino.

2.4.2 RASPBERRY PI

La Raspberry PI es una computadora de tamaño reducido (SBC) de bajo coste (\$35) cuyo tamaño es parecido al de una tarjeta de crédito y tiene unas características que lo hacen una opción muy potente para cualquier proyecto de electrónica, fue desarrollado en la universidad de Cambridge-Inglaterra, como una herramienta para enseñanza de programación y computación a los niños de ese país, la Raspberry PI (computing at school and BCS, 2012) tuvo un éxito enorme no solo en el mundo de la educación, sino también en el mundo de los desarrolladores de robótica, drones, internet de las cosas, domótica, etc. Siendo ahora la plataforma de desarrollo con una comunidad de desarrolladores muy grande haciendo esto mucho más fácil el aprendizaje a gente que no tiene experiencia con esta clase de dispositivos. Raspberry PI trabaja en un entorno Linux con el sistema operativo RASPBIAN que una es una modificación de DEBIAN.

2.4.2.1 Procesador

En la primera versión de Raspberry PI se tuvo al SoC broadcom BCM2835, con una arquitectura con un procesador ARM1176JZF a 700Mhz que era un procesador con arquitectura ARMv6, que fue usado en la primera generación de smartphones. Este chip incluía también un GPU videocore IV, y 256 MB de RAM.

La versión dos contaba con un chip BROADCOM BCM 2836 a 900MHz con un procesador ARM cortex-A7, de 32 bits y cuatro núcleos.

Y última versión que es la 3 cuenta con un chip BCM2837 a 1.2GHz con un procesador ARM Cortex-A53 1200 MHz.

2.4.2.2 Memoria RAM

La primera versión contaba con 256MB de RAM el cual era compartido entre el CPU y el GPU. En la versión dos contaba con 512MB RAM que se repartía entre el CPU y el GPU, y la versión 3 tiene 1GB de RAM, que lo hace mucho más potente y según el fabricante hasta 10 veces más potente que la primera versión.

2.4.2.3 Conectividad

Los modelos A, A+ y ZERO no cuentan con puerto Ethernet y si quieren conectarse a internet debían de usar un adaptador externo o un Dongle WIFI, en los modelos B y B+ se cuenta con un puerto ethernet usando el chip SMSC LAN9514, y la última versión cuenta además con WIFI 802.11n de 150Mbit/s y bluetooth 4.1 de 24mbit/s incorporados.

2.4.2.4 Periféricos

Las Raspberry PI pueden ser manejadas de la misma manera que una computadora común con un teclado y un ratón.

2.4.2.5 Reloj en tiempo real

Ningún modelo de Raspberry PI cuenta con un reloj en tiempo real, que es el encargado de conservar el tiempo mientras la computadora no está alimentada, la Raspberry pi normalmente obtiene la hora de un servidor NTP, sin embargo es posible agregarle un reloj en tiempo real a través de su bus I2C, entre los modelos de reloj más comunes están los chips DS3231 y DS1307.

2.4.2.6 Salidas y entradas de propósito general (GPIO)

Uno de las cosas más atractivas de Raspberry PI es que cuenta con salidas y entradas digitales de propósito general (GPIO, general purpose input output), pues a través de ellas la Raspberry pues puede comunicarse con el exterior, el GPIO en los modelos A y B constan de 26 pines y en el modelo B+ contiene 40 pines como muestra las imágenes siguientes.

GPIO#	2nd func.	Pin#	Pin#	2nd func.	GPIO#
	+3.3 V	1	2	+5 V	
2	SDA1 (I ² C)	3	4	+5 V	
3	SCL1 (I ² C)	5	6	GND	
4	GCLK	7	8	TXD0 (UART)	14
	GND	9	10	RXD0 (UART)	15
17	GEN0	11	12	GEN1	18
27	GEN2	13	14	GND	
22	GEN3	15	16	GEN4	23
	+3.3 V	17	18	GEN5	24
10	MOSI (SPI)	19	20	GND	
9	MISO (SPI)	21	22	GEN6	25
11	SCLK (SPI)	23	24	CE0_N (SPI)	8
	GND	25	26	CE1_N (SPI)	7
hasta aca los modelos A y B					
EEPROM	ID_SD	27	28	ID_SC	EEPROM
5	N/A	29	30	GND	
6	N/A	31	32		12
13	N/A	33	34	GND	
19	N/A	35	36	N/A	16
26	N/A	37	38	Digital IN	20
	GND	39	40	Digital OUT	21

Figura 14: Entradas y salidas de propósito general.

Fuente: Warner, T. (2014)

Como se puede ver en las Raspberry pi modelos A y B contamos con 26 pines de los cuales los pines 2 y 4 son de 5 v q sirven para alimentar algún dispositivo que se pueda conectar a la Raspberry PI, y el pin 1 es de 3.3v que tiene el mismo fin, y los

pinos 6, 9,14, 20, 25 son tierra, también cuenta con pinos de comunicación I2C, (pinos 3 y 5), y comunicación SPI por los pinos, 19, 21, 23, 24 y 26 con capacidad de comunicarse hasta con dos dispositivos que cuenten con ese BUS, también cuenta con pinos para comunicación serie a síncrona (pinos 8 y 10), el resto de pinos puede ser configurado como entrada o salida, algo que se tiene que tener muy en cuenta es que todos los pinos trabajan a 3.3 voltios por tanto lo que se vaya a conectar debe ser compatible con ese voltaje pues si se conectara algún dispositivo de 5 voltios terminaría dañando a la Raspberry PI.

2.4.3 SISTEMAS DE COMUNICACIÓN INALÁMBRICOS

Los sistemas de comunicación inalámbricos hacen uso del espectro electromagnético para la transmisión de información inalámbricamente, los sistemas de comunicación han evolucionado de manera importante en los últimos años y se ha hecho tan extenso que no es propósito de este trabajo estudiarlas todas, y de acuerdo a la naturaleza de nuestra investigación nos centraremos en 4 tecnologías que nos podrían ser útiles.

2.4.4 SISTEMAS DE COMUNICACIÓN POR SATÉLITE

Los satélites de comunicación pueden recibir y enviar desde el espacio ondas de radio en cualquier dirección que se tenga previsto en su diseño. Normalmente lo hacen desde y hacia grandes áreas de la Tierra, y en algunos casos también de y hacia otros satélites. El hecho de poder emitir desde un satélite de comunicación una señal que pueda recibirse con intensidad similar y simultáneamente en cualquier punto de una gran superficie geográfica es su característica más notable, y es la causa principal de su utilización, debido a sus implicaciones. La posibilidad de recibir señales emitidas por estaciones ubicadas en cualquier punto de una gran superficie de la Tierra es casi tan importante como la característica anterior, siendo indispensable para aprovechar los

satélites para servicios de comunicación bidireccional en toda su área de cobertura, es decir, para los que implican el diálogo entre los puntos extremos de la comunicación, como el servicio telefónico o un servicio interactivo de comunicación de datos. (Rosado, 2000)

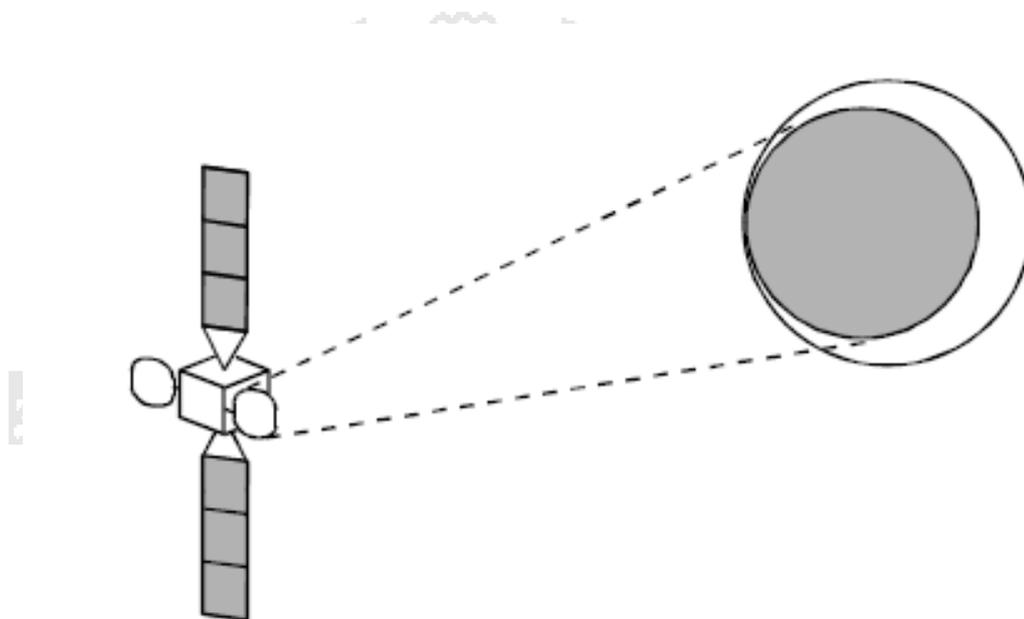


Figura 15: Satélite cubriendo más de un tercio de la superficie de la tierra.

Fuente: Rosado, (2000) recuperado de Comunicaciones por satélite, principios, tecnologías y sistemas.

La parte del espectro de radiofrecuencias atribuido por la UIT a la comunicación por satélite para cada uno de los tipos de servicio móvil por satélite (SMS), fijo por satélite (SFS), difusión (SRS), o entre satélites (SES), comprende porciones en el intervalo de aproximadamente 0.1 a 400 GHz. Actualmente, más del 90% de la capacidad de comunicación en órbita para fines comerciales se utiliza para el servicio fijo por satélite en la gama de frecuencias de 3.4 a 14.8 GHz (principalmente en las llamadas bandas C y Ku), considerando el número de satélites que las emplean y la reutilización de frecuencias en muchos de ellos. Para sus propios fines reglamentarios, la UIT considera tres regiones en el mundo: la región 1, que abarca África, los países Árabes, Europa y los países que anteriormente constituían la URSS; la región 2, que

abarca los países de América; y la región 3, que incluye a Asia y Oceanía, existiendo algunas diferencias menores en la atribución de frecuencias para cada región y excepciones registradas por países en forma individual. Por razones prácticas, a las bandas de frecuencias más comunes para el servicio por satélite se les designa por fabricantes de equipos, operadores de satélites y usuarios por medio de letras empleadas originalmente para radar, aunque no son utilizadas oficialmente por la UIT. De acuerdo con estas siglas, las principales bandas para los servicios por satélite son las mostradas en la **Figura 16** para la región 2 (Américas), como ejemplo.

Banda	Ejemplos de atribución (GHz) ⁺	Designación alternativa
L	1.525-1.71	Banda de 1.5 GHz
S	1.99- 2.20	Banda de 2 GHz
	2.5-2.69	Banda de 2.5 GHz
C	3.4-4.2, 4.5-4.8,	Banda de 4/6 GHz
	5.15- 5.25, 5.85-7.075	Banda de 5/7 GHz
X	7.2-8.4	Banda de 7/8 GHz
Ku	10.7-13.25, 13.75-14.8	Banda de 11/14 GHz, Banda de 12/14 GHz
Ka	27.0-31.0	Banda de 30 GHz

Figura 16: Distribución de frecuencias.

Fuente: Rosado, (2000), Comunicaciones por satélite, principios, tecnologías y sistemas.

2.4.5 REDES Y SERVICIOS

2.4.5.1 REDES

Una red de telecomunicación se establece con el propósito de transmitir y recibir señales de características específicas entre un conjunto determinado de estaciones distantes entre sí. Un sistema de satélites, o un satélite único, puede emplearse para permitir que se establezca a través de él una sola red o un número indeterminado de

redes de telecomunicación independientes. Los satélites de comunicación para servicios múltiples, y en buena medida los diseñados para un servicio específico, tienen la flexibilidad suficiente para permitir la operación de redes con características muy diversas, usualmente utilizando en forma exclusiva la capacidad de comunicación de los mismos en anchura de banda y potencia que requiera cada una de ellas, en una porción de la parte del espectro de frecuencias que tienen asignada. Por medio de los sistemas de satélites se pueden establecer servicios públicos completos hasta el usuario final, como es el caso de los de televisión directa a los hogares; servicios públicos apoyados parcialmente en redes terrenales de otros servicios, como el servicio móvil para flotillas de transportes terrestres; servicios privados totalmente independientes de otros servicios de telecomunicación, como en el caso de redes privadas de voz y datos; y servicios de apoyo a otros, proporcionados principalmente por medio de redes terrenales, como es el caso de la distribución de señales para televisión por cable y los enlaces de baja capacidad para telefonía pública en el medio rural.

2.4.5.2 Servicios

Entre las características de flexibilidad de los satélites destaca su capacidad de poder usarse para una amplia diversidad de servicios. Su aplicación inicial dominante durante una década fue para telefonía internacional y en pequeña proporción para tráfico de televisión internacional. Después, al desarrollarse los sistemas nacionales, se impulsó la diversificación de los servicios fijos al iniciarse los de datos y de distribución doméstica de señales de televisión, y casi al mismo tiempo se establecieron los sistemas de servicios móviles para aplicaciones marítimas. Actualmente el proceso de diversificación se está extendiendo aún más, con nuevos servicios a través de sistemas de satélites de usos múltiples y especializados, de órbitas geoestacionarias, medias y bajas, enlazados con estaciones terminales fijas, transportables, móviles y portátiles.

Trataremos brevemente algunas de las aplicaciones más usuales, clasificándolas principalmente por el tipo de señales originales que manejan, aunque esta forma de clasificación, como también otras, se hace cada vez menos conveniente por la tendencia a la digitalización y al uso de una misma onda portadora para transportar múltiples señales de distinta naturaleza en su origen.

2.4.5.2.1 Servicios de video

Los servicios de vídeo representan las aplicaciones de los satélites que han tenido mayor crecimiento en los últimos años, abarcando actividades de entretenimiento e información, negocios, educación y capacitación, y en forma incipiente en otras áreas. Inicialmente solo se transmitían por el sistema Intelsat señales de televisión para eventos de interés internacional, y aunque ha continuado, la demanda para esta aplicación, que es ocasional, ha tenido un incremento relativamente bajo. La mayoría de las señales de televisión para diversos usos se transmiten actualmente por sistemas regionales y nacionales para cobertura nacional a tiempo completo, y en los últimos años ha estado evolucionando una tendencia hacia el desarrollo acelerado de los servicios de televisión directa a los hogares, con cobertura nacional y regional. Estas señales, salvo circunstancias especiales, como en el caso de requerirse su recepción en zonas cercanas a los polos, se transmiten exclusivamente por medio de satélites geoestacionarios. Debido a sus múltiples ventajas, en los últimos pocos años se inició la digitalización generalizada de las señales de televisión, que se acentuará en forma acelerada con la relativamente reciente adopción de normas para su compresión. Como consecuencia de la importante reducción de la capacidad requerida para transmitir señales de televisión que se logra actualmente con la compresión digital, por su muy alta eficiencia, hubo temores de que se presentara una crisis de reducción de la demanda en los sistemas de satélites. No obstante, también se previó que la compresión

impulsaría los servicios que pueden prestarse por medio de señales de este tipo, y de hecho es lo que está ocurriendo, compensando la reducción en las anchuras de banda necesarias. Dentro de esta categoría existen sub servicios que se mencionaran a continuación:

- Radiodifusión
- Televisión por cable
- Educación a distancia
- Televisión directa

2.4.5.2.2 *Telefonía*

Los satélites también son muy usados en telefonía conectando lugares muy alejados e incluso telefonía internacional siendo estas manejadas por empresas privadas que también manejan la telefonía local, por otro lado cubre la telefonía nacional al comunicar lugares alejados o que están en zonas montañosas, donde no aún no ha llegado la red local de telefonía.

2.4.5.2.3 *Servicio de datos*

La aplicación de la transmisión de datos por medio de los sistemas de satélites ha originado una importante evolución tecnológica, especialmente en las estaciones terrenas, en relación con los métodos de acceso múltiple. Las primeras aplicaciones estaban orientadas a los servicios públicos de muy baja velocidad de transmisión y posteriormente a las redes privadas, para enlazar un equipo central de cómputo en la oficina matriz de una empresa con terminales en sus sucursales, las cuales tenían necesidades de velocidad de transmisión de datos que fueron creciendo progresivamente. Actualmente, con la tendencia a la digitalización de todas las señales, se están combinando y adaptando las tecnología empleadas anteriormente para cada tipo

de ellas en forma separada. Las características de una red para la transmisión de datos dependen de diversos factores tales como la configuración requerida, la velocidad de transmisión, el tráfico tanto individual como conjunto y el número de estaciones terminales previstos.

2.4.6 TELEFONIA CELULAR

O telefonía móvil es la comunicación inalámbrica aprovechando las ondas electromagnéticas utilizando un dispositivo llamado teléfono móvil que se conecta a una red de transceptores llamados células que conforman una red celular, por el cual se puede transmitir información de voz y acceso a internet de forma digital. (Huidobro, 2012)

2.4.6.1 REDES GSM

El comienzo de la comunicación móvil de forma digital apareció de la mano de la tecnología GSM (Global System for Mobile). Mediante ondas de radio se consiguió transportar información desde un terminal hasta otro. Su frecuencia inicial de transmisión fue 900MHz y posteriormente, 1800MHz. (en el Perú 850MHz y 1900MHz) Aunque actualmente es de 9,6Kbps por la escasez de su uso.

Con GSM se incluyó la utilización de tarjetas SIM (módulo de identidad del suscriptor) para la identificación del usuario y permitir así la conexión a la red móvil del operador en cuestión. Esto permitía el cambio entre móviles de forma rápida y con sólo tener que ingresar el código de acceso a la tarjeta SIM. No sólo apareció la tarjeta SIM, sino que también los SMS fueron posible gracia a esta tecnología. Aunque poco se hable de ellos en los tiempos que corren y en pleno auge de los servicios de mensajería instantáneos.

Uno de los principales problemas de esta tecnología era las limitaciones en cuanto a usuarios activos. Debido al rango de frecuencias de cada operadora y el hecho de que cada paso de información reservaba un canal, era un problema para gestionar grandes cantidades de tráfico. Esto propició la creación de las redes 3G.

La evolución a las redes 3G supuso al fin la posibilidad de navegar por internet desde un terminal móvil e inalámbricamente. Aquí tenemos que hablar de dos estándares que marcaron fuerte esta era de la tecnología móvil: GPRS y UMTS. Aclarar que del paso a las redes 2G a las redes 3G tenemos que hacer dos paradas, la primera en 2.5G (GPRS) y la otra, ya final, en 3G (UMTS).

2.4.6.2 Redes 3G

Tras este pequeño paso en las comunicaciones móviles, entramos en las redes 3G de la mano del estándar UMTS, basado en la tecnología W-CDMA (Wideband Code Division Multiple Access). Gracias a esta tecnología se podían tener varios usuarios transmitiendo información simultáneamente y tasas de 12,2Kbps para voz y hasta 384kbps para datos. Visto toda la historia de las redes móviles hasta las redes 3G, ahora pasamos a hablar detenidamente de las redes entre 3G y 4G. Por supuesto, también hablaremos de redes 4G.

Cada vez se buscaban mejores tasas de envío y recepción de información a través de terminales móviles. Es por ello que las redes móviles siguieron evolucionando y salió a luz HSPA (High-Speed Packet Access). Aquí se combinan las tecnologías 3G y posteriores, es decir contamos con la unión del HSDPA (3.5G) y HSUPA (3.75G).

Esta tecnología permitía velocidades de hasta 14,4Mbit/s de bajada y hasta 2Mbit/s de subida. Aunque claro, todo esto dependía del estado de la red al momento de usarla. Pero del HSPA pasamos a una versión mejorada de estas redes, hablamos de

HSPA+ o Evolved HSPA. Esta evolución supuso un salto considerable en cuanto a las velocidades de subida y bajada de información. Mediante tecnologías multi-antena (MIMO) y modulación 64-QAM se podían conseguir velocidades de hasta 84 Mbps de bajada y 22 Mbps de subida. Aunque claro, esto son picos alcanzados, ya que la mayoría del tiempo rondaremos las velocidades de HSDPA y HSUPA. En caso de que se utilicen canales con anchos de banda superiores a los 5 MHz sí que se podrían conseguir velocidades cercanas a las que dijimos antes.

2.4.6.3 Redes 4G

Actualmente estamos en la era de las redes 4G. Una evolución considerable de las redes móviles con la que podremos disfrutar de altas velocidades de transmisión de información de forma completamente inalámbrica. ¿Recordáis al comienzo las tasas de conexión?, comparadla con las que existen en 4G y veréis el salto tan enorme (y rápido) que es. Esta nueva era de redes móviles se estableció tras un comité de la UIT (Unión Internacional de Telecomunicaciones). Aquí se establecían los nuevos niveles de velocidades de transmisión de información. Estos niveles son de 100 Mbit/s para terminales en movimiento y hasta 1Gbit/s para terminales en reposo.

Las redes 4G están basadas completamente en el protocolo IP ya que se unen redes de cable e inalámbricas. De este modo es más eficaz la transferencia de información y se pueden alcanzar los niveles establecidos por la UIT. También se incluyeron técnicas de rendimiento avanzado de radio como es MIMO y OFDM.

Sobre la red LTE hay que decir que no es una red puramente 4G. Esto es porque no cumple los requisitos establecidos por la UIT para que pueda llamarse red 4G. Esto es debido a que no alcanza los valores pico requeridos por las redes 4G y tampoco, la eficiencia espectral de las ondas de radio que utiliza.

2.5 FTP (FILE TRANSFER PROTOCOL, PROTOCOLO DE TRANSFERENCIA DE FICHEROS)

Es un servicio que los usuarios lo utilizan frecuentemente (descarga de drivers, música, documentos, etc.), pero esto es sólo una parte del servicio, ya que también es posible, implementar en nuestra máquina, un servidor FTP para que otros usuarios se puedan conectar a nuestra computadora y recoger/dejar información en una zona concreta. Existen dos tipos de transferencias:

- **Descarga (download).** Consiste en traer un archivo a nuestro ordenador desde un servidor remoto. También se dice "*bajar un archivo*".
- **Carga (upload).** Consiste en llevar un archivo desde nuestro ordenador a un servidor. También se dice "*subir un archivo*".

El **servicio FTP** (*File Transfer Protocol*), es por tanto un servicio que se utiliza para transferir información, almacenada en ficheros, de una máquina remota a otra local, o viceversa. Para poder realizar esta operación es necesario conocer la **dirección IP** (o el "**nombre**") de la máquina a la que nos queremos conectar para realizar algún tipo de transferencia.

Es fundamental distinguir entre **máquina local** y **máquina remota**:

- **MAQUINA LOCAL:** Es aquella desde donde nos conectamos para hacer la transferencia, es decir, donde ejecutamos ftp.
- **MAQUINA REMOTA:** Es aquella a la que nos conectamos para transferir información.

El **servicio FTP** se puede realizar a través de los navegadores de Internet (**Netscape** e **Internet Explorer**), así como a través de **clientes FTP**.

Los **servidores FTP** controlan el acceso de los usuarios a su sistema de carpetas. Esto quiere decir que, para conectarse a un servidor FTP, necesitamos un **usuario (login)** y una **contraseña (password)**. Se ofrece una alternativa de acceso no autenticado: El usuario *anonymous*, que no lleva contraseña. Normalmente, este usuario tiene los permisos restringidos.

2.5.1 SERVIDORES FTP

El **servidor FTP** es el software que tienen instalado las máquina que ofrecen servicio de FTP a la Comunidad Internet. El servidor de archivos debe admitir las transferencias de tipo FTP, por lo que deberá ser un ordenador especialmente preparado para esta tarea.

Los **servidores FTP de Internet** son en su inmensa mayoría sistemas **UNIX/LINUX** aunque se puede montar un servidor FTP en nuestro ordenador. Para ello, necesitaremos un programa específico; hay varios muy populares, gratuitos, algunos incluso en castellano.

Los **servidores FTP** controlan el acceso de los usuarios dentro de su sistema de carpetas. Esto quiere decir que, para conectarse a un servidor FTP, necesitamos un **usuario (login)** y una **contraseña (password)**. Se ofrece una alternativa de acceso no autenticado: el usuario *anonymous*, que no lleva contraseña. Normalmente, este usuario tiene los permisos restringidos. No obstante, **ES RESPONSABILIDAD DEL PROPIETARIO DEL SERVIDOR FTP, LA POLÍTICA DE CREACIÓN Y MANTENIMIENTO DE USUARIOS Y DIRECCIONES IP. El Servicio de Informática, recomienda ENCARECIDAMENTE, QUE SE ELIMINE/DESHABILITE el usuario anonymous, y que se creen los usuarios autenticados con login y password en el caso de sé monte un servidor FTP con el software suministrado desde nuestro Centro de Descarga.**

2.5.2 CLIENTES FTP

El acceso convencional a los **servidores FTP** se realiza por medio de **programas clientes específicos para el protocolo FTP**, que permiten cosas como la transferencia de grupos de ficheros o directorios completos. Es la forma más fácil de transferir archivos entre un PC y un servidor remoto, por tanto, para ficheros grandes es mejor utilizar **Clientes FTP** que el correo electrónico. Los **clientes FTP** más utilizados son:

- **En Microsoft Windows: WS-FTP** (cliente utilizado en la Universidad de Jaén) y **CuteFTP**.
- En Linux: **ncftp** (en modo carácter) y **gFTP** (gráfico).

2.6 REDES VPN

Una VPN o Red Privada Virtual no es más que una extensión de la red local de una entidad, a la que se le agregan unas configuraciones y componentes (hardware y software) que le permitan incorporarse sobre una red de recursos de carácter público como Internet, ATM o FrameRelay, pero manteniendo un entorno de carácter confidencial y privado que posibilite trabajar al usuario como si estuviera en su misma red local. La comunicación entre los dos extremos de la red privada a través de la red pública se hace creando túneles virtuales entre esos dos puntos y usando sistemas de encriptación y autenticación que aseguren la confidencialidad e integridad de los datos transmitidos a través de esa red pública.

2.6.1 VENTAJAS DE UNA VPN PARA UNA ORGANIZACIÓN

- Extensiones de conectividad a nivel geográfico.
- Mejoras de la seguridad.
- Reduce costos al ser instalado frente a las redes WAN más utilizadas.

- Mejora la productividad.
- Simplifica la o topología de red.
- Proporciona oportunidades de comunicaciones adicionales.

2.7 HIPOTESIS DE LA INVESTIGACION

2.7.1 Hipótesis general

El prototipo nos permite obtener información útil para el monitoreo del volcán Ubinas.

2.7.2 Hipótesis específicas

H.1. El hardware necesario para implementar el dispositivo de medición de potencial espontaneo será de bajo coste y su implementación será rápida.

H.2. La tecnología de telecomunicaciones que se usará será de bajo coste y de fácil implementación.

2.8 OPERACIONALIZACION DE VARIABLES

2.8.1 Variable independiente

Es el hardware y la tecnología de telecomunicaciones que se usara para la implementación del prototipo de medición de potencial espontaneo.

2.8.2 Variables dependientes

- La utilidad del prototipo para el monitoreo volcánico.
- El tiempo de muestreo.
- La precisión y exactitud de las muestras de voltaje tomadas por el prototipo.
- Velocidad de transmisión de datos tomados por el prototipo.



CAPITULO III
MARCO METODOLOGICO

3.1 DISEÑO, NIVEL Y TIPO DE LA INVESTIGACION

3.1.1 Diseño de la investigación

El diseño de la investigación es el plan o estrategia que el investigador deberá seguir para responder las preguntas planteadas por la investigación (Sabino, 1992). Bajo este enfoque la presente investigación es **experimental y descriptiva**, experimental porque para responder la pregunta principal es necesario someter el prototipo a un experimento bajo condiciones reales en campo, y descriptiva porque para lograr los objetivos específicos es necesario conocer los aspectos más importantes de las tecnologías de las que podemos disponer para la conclusión del prototipo.

3.1.2 Nivel de la investigación

El nivel de investigación se refiere a la profundidad del conocimiento que se busca lograr con la investigación, por tanto el nivel de la presente investigación es **perceptual**, porque no se pretende tener un conocimiento profundo de las distintas partes que componen la presente investigación, y en cierto modo es **exploratoria** pues (Baptista Lucio , Hernadez Sampieri, & Fernandez Collado, 2006) señalan que las investigaciones exploratorias buscan abrir nuevos caminos en el desarrollo del conocimiento humano. Y la presente investigación siendo un prototipo busca abrir un camino para un nuevo método de monitoreo volcánico.

3.1.3 Tipo de investigación

El tipo de investigación se refiere al alcance de la investigación el cual depende del enfoque que se le quiera dar a la investigación y el conocimiento que ya se tenga respecto al tema, bajo esta óptica la presente investigación es de tipo **aplicada** pues busca resolver en cierta medida el problema de monitoreo volcánico proveyendo una nueva variable a corto plazo.

3.2 POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN

3.2.1 Población.

Siendo la población el conjunto de las entidades, o cosas respecto a las cuales se basa las conclusiones de una investigación, para nuestro caso la población está definida por las variaciones en el potencial espontaneo en el edificio volcánico de volcán Ubinas en un tiempo indeterminado.

3.2.2 Muestra.

Se define la muestra como parte que se estudia y es representativa de la población, es decir un segmento que tiene las características y propiedades de la población, por tanto nuestra muestra serán la medición de las variaciones del potencial espontaneo en un periodo de tiempo determinado.

3.3 UBICACIÓN Y DESCRIPCION DE LA INVESTIGACIÓN

3.3.1 Ubicación

La investigación descriptiva, que contempla la determinación del hardware y las tecnologías de transmisión de datos, se realizaron en las instalaciones del observatorio vulcanológico del INGEMMET (OVI), ubicado en la ciudad de Arequipa. Ver **Figura 17**.

Y la parte experimental de la investigación se realizó en el flanco sur del volcán Ubinas en las coordenadas latitud $16^{\circ}21'18.26''S$, longitud $70^{\circ}53'7.96''O$, a una altura respecto al mar de 4800msnm aproximadamente. Perteneciente al distrito de Ubinas, provincia General Sánchez Cerro, en el departamento de Moquegua, ver **Figura 18**.

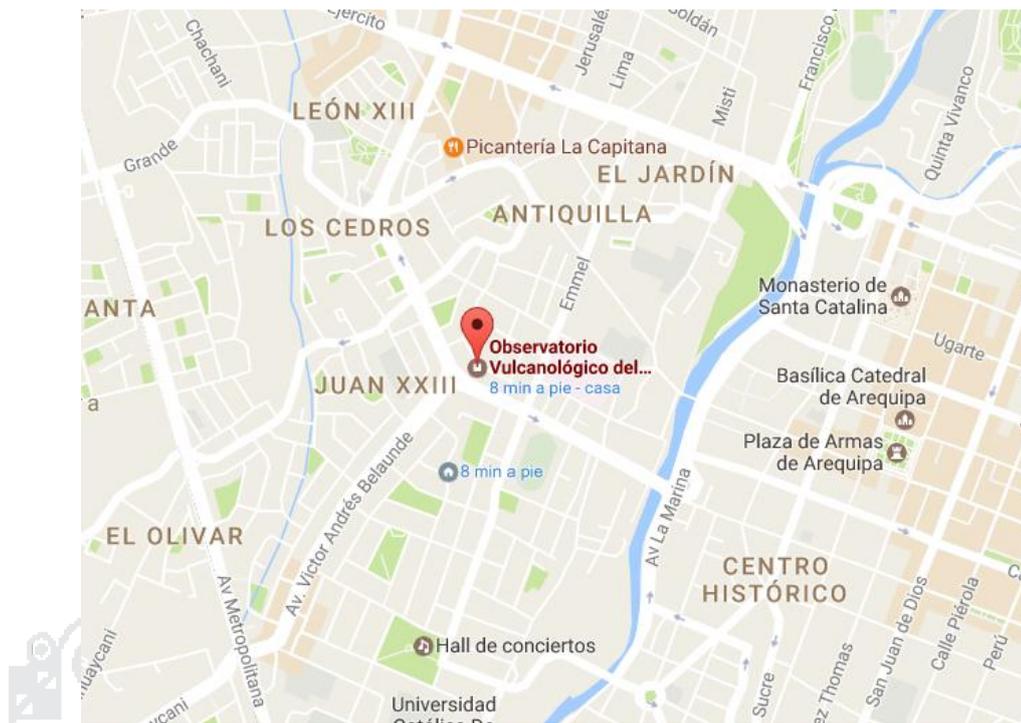


Figura 17: Ubicación del observatorio vulcanológico del INGEMMET.

Fuente: Googlemaps (2016).



Figura 18: Ubicación del punto de medición de potencial espontaneo.

Fuente: GoogleEarth, (2016).

3.3.2 Descripción de la investigación

La presente investigación pretende demostrar la utilidad de un prototipo en una técnica en proceso de desarrollo para monitoreo de volcanes, el investigación consta de 4 fases que se describen a continuación.

- **Fase 1:** Esta fase trata esencialmente búsqueda de información, consulta bibliográfica y sitios web, sobre plataformas de desarrollo, elección de componentes y técnicas o métodos de envío de datos inalámbricamente.
- **Fase2:** Esta fase comprende el diseño e implementación del prototipo, calibración en laboratorio y pruebas preliminares.
- **Fase 3:** Comprende la instalación del prototipo en campo, en el lugar designado por el profesional en geofísica.
- **Fase 4:** Comprende la obtención y el proceso de verificación, que consiste en correlacionar las variaciones de estos datos con datos obtenidos de los otros métodos de monitoreo volcánico.

3.4 TECNICAS E INSTRUMENTOS DE RECOLECCION DE DATOS

La recolección de datos se refiere a como y que medios se usan para la obtención de la información que serán de utilidad para la corroboración de nuestras hipótesis, por lo tanto resumimos este apartado en la **Tabla 1**:

Tabla 1: Técnicas e instrumentos de investigación.

Técnicas	Instrumentos
Consultas bibliográficas y de bases de datos.	Papers, internet, Bases de datos de otros métodos.
Mediciones directas.	Instrumentos de medición (multímetro).

Elaboración: Propia.

3.5 TECNICAS PARA EL PROCESAMIENTO Y ANALISIS DE DATOS

El procesamiento de los datos obtenidos por el prototipo se basara en la **CORRELACION** con los datos de otros métodos de monitoreo volcánico, (visual, sísmica), comparando las variaciones respecto al tiempo.

3.6 DESCRIPCION DEL PROCESOS DE MEDICIÓN DEL POTENCIAL ESPONTANEO.

Como de se indicó en el capítulo II el potencial espontaneo es el voltaje natural del suelo que depende, entre otras cosas de los cambios de temperatura interna del edificio volcánico debido al ascenso de magma, por tanto la medición de potencial se hará en uno de los flancos del volcán, una vez identificado la zona donde se hará la medición se enterrara cada electrodo a una profundidad de 70cm con una separación de 100 metros entre electrodos, de manera radial con respecto al cráter del volcán, esta separación puede variar dependiendo del terreno y los estudios previos que se hubieren hecho. Los electrodos estarán conectados al prototipo mediante un cable 14 AWG para reducir en lo posible la resistencia del cable. El periodo de muestreo es de 1 segundo y los datos son enviados hasta la sede del OVI mediante una VPN (red privada virtual), por internet móvil usando un modem USB (dongle).

3.6.1 Requerimientos básicos para la medición el potencial espontaneo que ha de cumplir el prototipo

- Impedancia de entrada $\geq 10 \text{ M}\Omega$.
- Resolución $\pm 1\text{mV}$ mínimo.
- periodo mínimo 1s, programable.
- Rango de entrada programable.
- Almacenamiento de datos en memoria interna.

- Con capacidad de enviar datos periódicamente vía puerto ethernet.
- Con capacidad de enviar datos vía internet usando un modem USB GPRS/3G.
- Bajo consumo.

3.7 ELECCION DE TARJETA DE DESARROLLO

Para la elección de la tarjeta de desarrollo se tomaron varios criterios que se resumen en la **Tabla 3** sin embargo la razón por las que se tomaron en cuenta solo esas tres tarjetas de desarrollo es porque son las que podemos encontrar en el mercado local, existen otras tarjetas de desarrollo similares en potencia a la Raspberry PI 3 B+ pero no es posible encontrarlas en el mercado local, para obtener una de esas se deben comprar por internet y él envío tarda al menos un mes y cuando se trabaja con prototipos siempre hay riesgo de dañar componentes, si sucediera ese caso significaría mucho tiempo de retraso, sin mencionar que son algo más caras que la Raspberry PI B+, además en ambos casos la comunidad de desarrolladores es extensa y activa.

Tabla 2: Resumen de eleccion de tarjeta de desarrollo.

Tarjeta de desarrollo	Especificaciones							Lenguajes de programación	Comunidad	Ampliable	Componentes del prototipo	Costo	
	CPU	ROM	RAM	ADC	Puertos USB	Entradas y salidas	Unid S/.					Tot al S/.	
Raspberry PI 3 B+	ARMv8 1.2Ghz	SD	1GB	NO	4	40 Pines		Python, C/C++, etc	Muy Activa	Ampliable	Módulo ADC ad1115	25	320
											RTC ds3231	15	
											Modem USB mfi180	50	
											Regulador de voltaje	10	
											Raspberry PI 3B+	220	
Arduino DUE	AT91SAM 3X8E 84 MHz	512 KB	96 KB	12	1	54 Pines	C,	Activa	Limitado	Módulo ADC ad1115	25	490	
										RTC ds3231	15		
										Shield ethernet	70		
										Shield GSM	150		
										Arduino DUE R3	230		
Arduino Mega 2560	ATmega2560 16 MHz	256 KB	8KB	16	NO	54 Pines	C,	Muy activa	Limitado	Módulo ADC ad1115	25	350	
										RTC ds3231	15		
										Shield ethernet	70		
										Shield GSM	150		
										Arduino Mega 2560 R3	90		

Elaboracion: Propia

3.8 ELECCION DE LA TECNOLOGIA DE TELECOMUNICACIONES

El criterio para la selección de la tecnología y servicio de telecomunicaciones está basado en aspectos más superficiales, como la disponibilidad del servicio, costo del servicio y el equipamiento e instalación, logística necesaria, mas no nos adentramos en temas técnicos propios de cada sistema como el ancho de banda o velocidad de transmisión, configuraciones avanzadas, etc. Pues los datos que se transmitirán son muy pequeños y no se requeriría de un medio que soporte un gran tráfico de datos, además tomamos en cuenta que estas tecnologías ya están establecidas por tanto no es intención de esta investigación adentrarse en ese ámbito. Por tanto como muestra la **Tabla 3** es evidente la elección de una red GSM/3G, pues su instalación no requería de una cantidad grande de recursos ni estos serían caros y el servicio resulta económico en comparación a las otras opciones , además solo necesitaría de dos personas como personal, y debido al poco costo la perdida por hurto no es importante, y el tiempo que requeriría seria de un día, cosa contraria pasa si nos fuéramos por la opción satelital, pues al ser más caro el equipamiento la perdida por hurto seria significativa, por lo que habría que pensar en una instalación más segura lo que implica más costos en materiales, personal y varios días de trabajo, del mismo modo sucede si buscáramos conectarnos a la red que tiene el OVI que por desgracia desde el punto de las mediciones no enlaza con ninguna de las estaciones que tiene su red, pero si en caso habría enlace, los costos se instalación serian similares a si instaláramos la opción satelital, cabe mencionar también que estas dos últimas opciones requerirían de protección con rayos, es decir instalar una puesta a tierra y eso incrementaría aún más los costos de instalación.

Tabla 3: Tabla resumen de elección de tecnología de telecomunicación

Tecnología de telecomunicación	Operador a	Cobertura	Equipamiento	Logística	Costos	
					Instalación	Servicio S/.
Red GSM/3G	Claro	Si	Modem	-Poco personal. -Sin estructura	Muy bajo costo	Desde 29.9
	Movistar	No	Modem			
Red Satelital	Movistar	Si	-Antena parabólica -IDU	-Mucho personal -Estructura y concreto	Muy costoso	Desde 150
	Viasatelite	Si	-Antena parabólica -IDU	-Mucho personal -Estructura y concreto	Muy costoso	Desde 300
		Si	-Antena parabólica -IDU	-Mucho personal -Estructura y concreto	Muy costoso	
Radio enlace dedicado	Red propia	No	Radio ethernet 900MHz freewave	-Mucho personal -Concreto, estructura	Muy costoso	

Elaboración: Propia.

3.9 DESCRIPCION DE COMPONENTES

El diseño del prototipo se enfoca en dos aspectos principales que son el hardware o parte física del prototipo y el software que es la parte lógica del sistema, y la selección y construcción de ambas responden a las necesidades que implican la medida, del

potencial espontaneo, el almacenamiento de los datos y la posterior transferencia de información. Por tanto nos enfocaremos en esos dos aspectos.

3.9.1 Determinación del hardware para el prototipo

Para definir el hardware que se va a usar vale definir primero los bloques que compondrán el prototipo definiendo las tareas específicas de cada bloque el cual se resume en la **Figura 19**, donde el ADC es el encargado de digitalizar un valor analógico en este caso el valor de potencial espontaneo, al mismo tiempo debe cumplir con los requerimientos de impedancia de entrada, resolución, que requiere este proceso, el reloj debe mantener la hora independientemente de la alimentación del sistema y provee la hora para sincronizar los diferentes procesos que hará el procesador, como leer datos del ADC, enviar los datos y hacer reinicios de rutina del sistema. Y el procesador será el corazón de todo realizando las tareas de manejar el ADC, organizar y almacenar las lecturas y conectarse a una red a través del modem USB GPRS/3G para enviar los datos, y el modem será el medio físico por que el sistema se conectara a la red celular GPRS/3G y por ende a internet y a través de una red privada virtual (VPN), se conectara a la red del OVI.

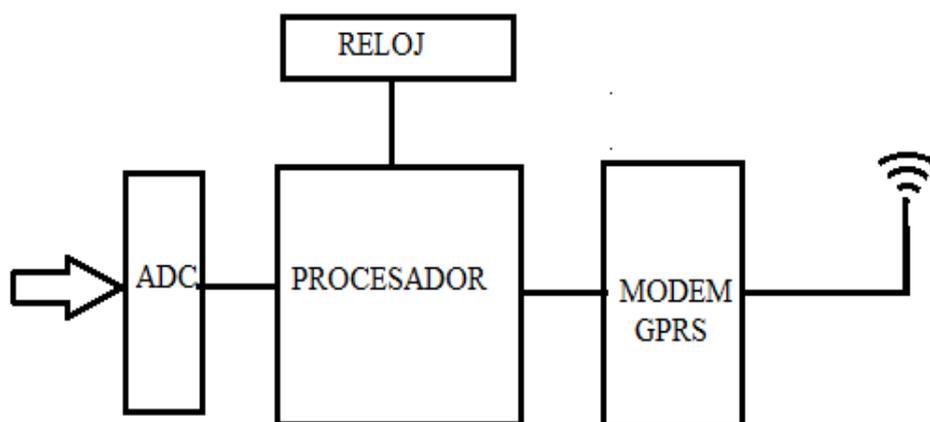


Figura 19: Diagrama de bloques del prototipo.

Fuente: Elaboración propia.

3.9.2 Módulo ADS1115

El ADS1115 es un conversor analógico digital delta-sigma de 16 bits, de ultra bajo consumo y compatible con interfaz I^2C , contenido en un empaquetado pequeño, pero es posible conseguirla en forma de modulo en una pequeña placa listo para conectar. Incorpora un voltaje de referencia con muy bajo risis y un oscilador, pero lo que hace más interesante a este ADC es que incorpora un amplificador de ganancia programable o PGA con una impedancia de entrada alta, lo cual permite la conexión directa con una variedad de sensores, por tanto no requiere mayor adaptación de señal a la entrada del ADC, puede llegar a un data rate de hasta 860 muestras por segundo (SPS). Y el amplificador de ganancia programable admite una entrada de $\pm 256 mV$ hasta $\pm 6.144 V$ permitiendo medidas precisas de señales grandes y pequeñas, también incorpora un multiplexor que permite que se puedan tener 2 entradas diferenciales o 4 entradas simples de voltaje y usa un comparador digital usado para detección de altos y bajos voltajes. El ADS1115 puede trabajar de dos modos de conversión, el modo continuo de muestreo o solo una muestra cuando se requiera y si esta en modo de una sola muestra el dispositivo entra en modo de bajo consumo reduciendo mucho más el consumo del dispositivo. (Industries, 2016)

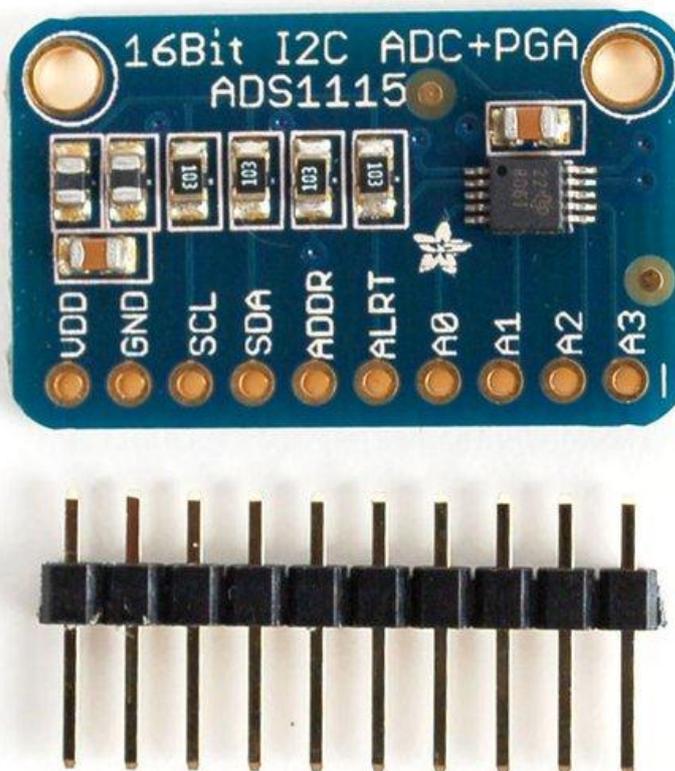


Figura 20: Modulo ADC ADS 1115.

Fuente: Earl, B. (2016).

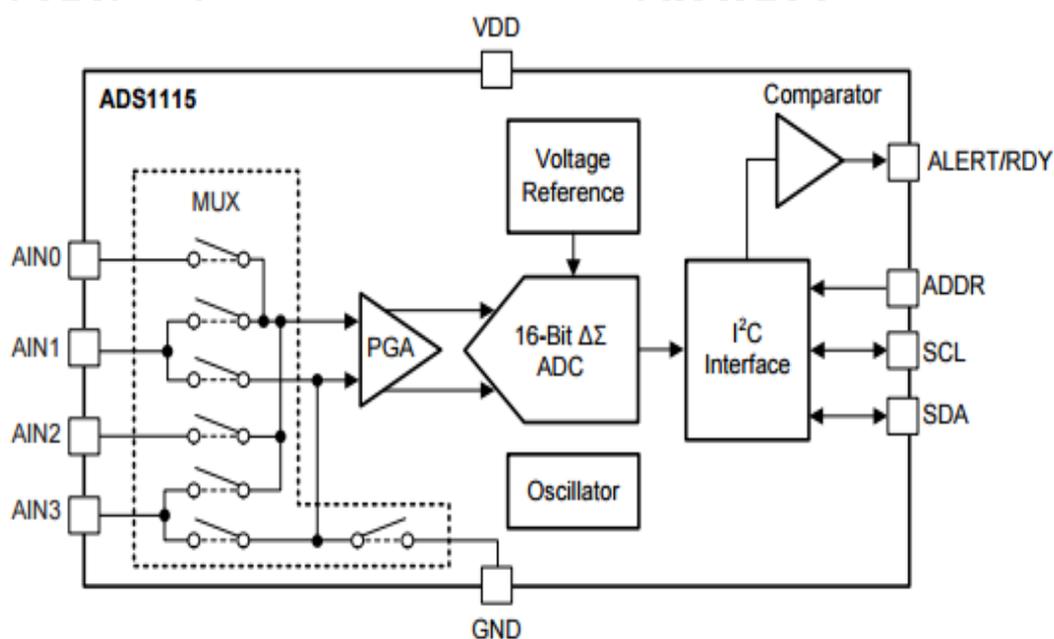


Figura 21: Diagrama de bloques del ADS1115.

Fuente: Earl, B. (2016).

3.9.2.1 Características

- Resolución: 16bits.
- Velocidad de muestreo programable: de 8 a 860 SPS.
- Voltaje de alimentación y niveles lógicos: de 2 a 5.5v.
- Bajo consumo: en modo continuo 150uA, y en modo muestra simple entra en modo de bajo consumo.
- Voltaje de referencia interno de bajo rizado.
- Oscilador interno.
- Amplificador de ganancia programable hasta x16.
- Interface I2C hasta 4 posibles direcciones configurado con pines.
- 4 entradas simples o 2 entradas diferenciales, configurable.
- Comparador programable.

3.9.2.2 Impedancia de entrada diferencial

La impedancia de entrada es uno de los parámetros más importante a tomar en cuenta en nuestro prototipo pues se trata de medir señales de muy bajo valor y cuya fuente puede ser de una impedancia variable dependiendo de factores como la composición del suelo, como se puede ver en la **Tabla 4** la impedancia de entrada se reduce mientras más se reduce el rango del amplificador programable.

Tabla 4: Impedancia de entrada según el rango de entrada.

FS (V)	DIFFERENTIAL INPUT IMPEDANCE
$\pm 6.144V^{(1)}$	22M Ω
$\pm 4.096V^{(1)}$	15M Ω
$\pm 2.048V$	4.9M Ω
$\pm 1.024V$	2.4M Ω
$\pm 0.512V$	710k Ω
$\pm 0.256V$	710k Ω

Fuente: Earl, B. (2016).

3.9.2.3 Rango de entrada

El rango de entrada está definido por el amplificador de ganancia programable PGA y esta implementado justo antes del conversor sigma-delta la **Tabla 5** muestra los rangos de voltaje de entrada admitidos por el PGA, además del voltaje por bit.

$$LSB = \frac{FSR}{2^{16}} \tag{4}$$

Tabla 5: Rangos de entrada del PGA.

FSR	LSB SIZE
±6.144 V ⁽¹⁾	187.5 μV
±4.096 V ⁽¹⁾	125 μV
±2.048 V	62.5 μV
±1.024 V	31.25 μV
±0.512 V	15.625 μV
±0.256 V	7.8125 μV

Fuente: Earl, B. (2016).

Tabla 6: Configuración de PGA y su correspondiente rango de entrada.

PGA SETTING	FS (V)
2/3	±6.144V ⁽¹⁾
1	±4.096V ⁽¹⁾
2	±2.048V
4	±1.024V
8	±0.512V
16	±0.256V

Fuente: Earl, B. (2016).

3.9.2.4 Programación del módulo ADC ADS1115

El módulo ADC se programa a través de una interfaz I2C, y puede soportar múltiples dispositivos que también cuenten con esta interfaz pero estos pines han de tener una resistencia de pull-up que haga que el estado de los pines mientras no esté activo sea alto, la comunicación entre dos dispositivos I2C se lleva a cabo siempre que uno actúe como esclavo y el otro como maestro, en este caso este dispositivo será

siempre escalado, dependiendo siempre del maestro para que pueda transmitir la información que contenga en sus registros, el Bus I2C consiste en dos pines que son el SCL y SDA, el SCL se encarga de proveer la señal de reloj y SDA se encarga de transportar los datos bit a bit con cada pulso de SCL.

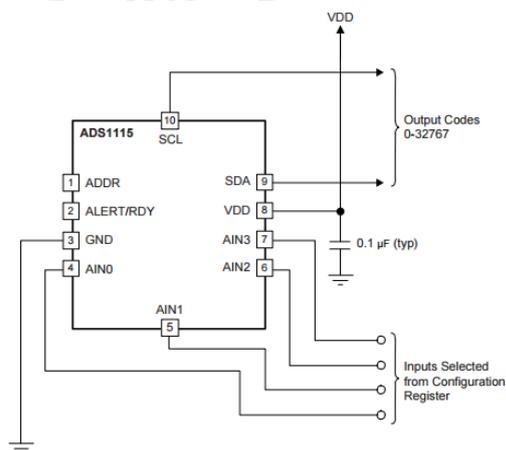


Figura 22: ADS1115.

Fuente: Recuperado de hoja de datos del ADS1115.

La dirección está definida por el pin ADDR y la dirección dependerá donde vaya conectado, se pueden establecer hasta 4 direcciones diferentes lo cual quiere decir que se pueden conectar hasta 4 ADS1115 en un mismo maestro como muestra la **Figura 23** las direcciones están indicadas en la **Tabla 7**.

Tabla 7: Conexión del pin ADDR y la correspondiente dirección.

ADDR PIN CONNECTION	SLAVE ADDRESS
GND	1001000
VDD	1001001
SDA	1001010
SCL	1001011

Fuente: Recuperado de hoja de datos del ADS1115.

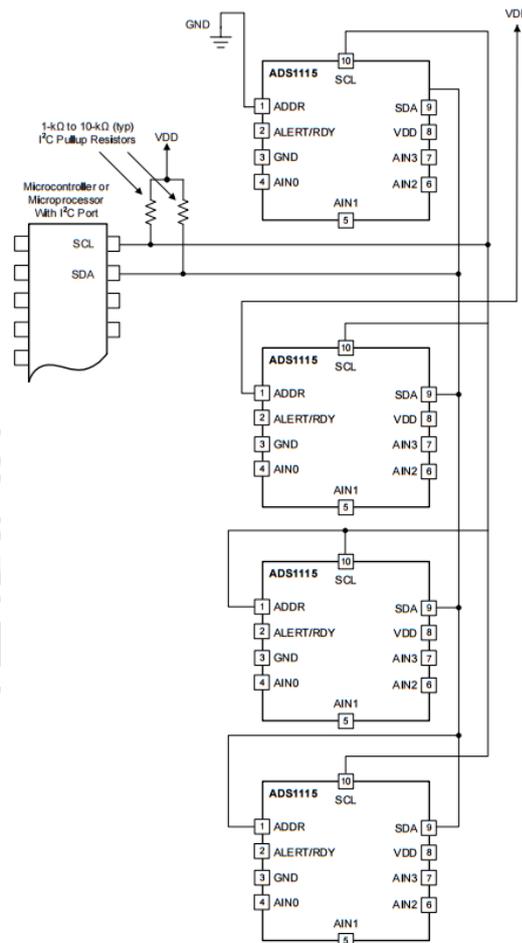


Figura 23 ADS1115 conectados a un solo maestro.

Fuente: Recuperado de hoja de datos del ADS1115

3.9.3 Reloj en tiempo real DS3231

Es un reloj en tiempo real muy preciso y de bajo costo que cuenta con un oscilador compensado en temperatura lo que garantiza una mínima desviación en la medida del tiempo a lo largo de periodos largos de tiempo, actualmente es posible encontrar este chip en una pequeña placa que incluye un receptáculo para una batería CR2032, que conserva el tiempo aun después de que se le quite la alimentación principal.

Este reloj es capaz de mantener segundos, minutos, horas, días, e información del año incluyendo años bisiestos, los datos son transferidos mediante un bus I2C

bidireccional por el que también es posible corregir la hora. También cuenta con un pin de alarma que puede ser configurado que se active cuando se igualan los segundos, segundos y minutos, segundos, minutos y horas, segundos minutos, horas y días, etc. Este pin puede utilizarse para utilizar interrupciones, sincronizar tareas, etc.

3.9.3.1 Características

- Voltaje de alimentación: 2.3v a 5.5v.
- Voltaje de batería de respaldo: 3v.
- Bus de comunicación: I2C bidireccional.
- Pin de alarma: sí.
- Temperatura de trabajo: -40°C a 85°C.
- Desviación de tiempo: ± 2 minutos por año.



Figura 24: Modulo Reloj DS3231.

Fuente: Ada,(2016).

3.9.3.2 Mapa de registros

Tabla 8: Mapa de registro del DS3231.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00-59
01h	0	10 Minutes			Minutes				Minutes	00-59
02h	0	12/24	AM/PM 20 Hour	10 Hour	Hour				Hours	1-12 + AM/PM 00-23
03h	0	0	0	0	0	Day			Day	1-7
04h	0	0	10 Date		Date				Date	01-31
05h	Century	0	0	10 Month	Month				Month/ Century	01-12 + Century
06h	10 Year			Year				Year	Year	00-99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00-59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00-59
09h	A1M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 1 Hours	1-12 + AM/PM 00-23
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1-7
					Date				Alarm 1 Date	1-31
0Bh	A2M2	10 Minutes			Minutes				Alarm 2 Minutes	00-59
0Ch	A2M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 2 Hours	1-12 + AM/PM 00-23
0Dh	A2M4	DY/DT	10 Date		Day				Alarm 2 Day	1-7
					Date				Alarm 2 Date	1-31
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Fuente: Recuperado de hoja de datos del ds3231.

3.9.4 Modem USB HSDPA modelo ZTE MF180

El modem ZTE MF180 es un modem USB inalámbrico multi-modo 3G compatible con redes HSDPA/UMTS/EDGE/GPRS/GSM se puede conectar mediante USB a cualquier computadora sea portátil o de escritorio y haciendo uso de un software manager se puede conectar a internet a través de una red móvil HSDPA/UMTS/EDGE/GPRS/GSM, también puede usarse como un lector de memoria SD.

3.9.4.1 Especificaciones

- Interface: USB.
- Certificaciones: FC/FCC.
- Velocidad de transmisión: 3.6MB/s.

- Bandas de frecuencia 3G: HSDPA/UMTS 2100MHz.
- Bandas de frecuencia 2G: GSM/GPRS/EDGE 850/900/1800/1900MHz.
- Temperatura de operación: -10C - +55C.
- Lector de memoria SD: 8GB.



Figura 25: Modem USB ZTE MF180.

Elaboración: Propia.

3.9.4.2 Tarjeta SIM

Una tarjeta SIM (acrónimo en inglés de subscriber identity module, en español módulo de identificación de abonado) es una tarjeta inteligente desmontable usada en teléfonos móviles y módems HSPA o LTE que se conectan al puerto USB. Las tarjetas SIM almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la suscripción del cliente de un terminal a otro simplemente cambiando la tarjeta. Cabe mencionar que la operadora a la que corresponda la tarjeta SIM dependerá de la cobertura existente en el lugar donde se pretenda instalar el prototipo, en nuestro caso elegimos la operadora CLARO, pues es la única que tiene cobertura en la zona donde se pretende instalar el prototipo.



Figura 26: Tarjeta SIM de operadora claro.

Elaboración: Propia

3.9.5 Plataforma de desarrollo: Raspberry pi 3 B+

Para la realización de este prototipo utilizamos la Raspberry PI 3 B+ pues es la más nueva versión que adiciona nuevas características, además su costo es igual al de versiones anteriores como la Raspberry PI 2 B+, además cuenta con características adicionales que lo hace más fácil de acceder a ella a través de un secure Shell (ssh), las características más resaltantes de la Raspberry pi 3 B+ se mencionan a continuación:

Nuevas características:

- Procesador a 1,2 GHz de 64 bits con cuatro núcleos ARMv8.
- 802.11n Wireless LAN.
- Bluetooth 4.1.
- Bluetooth Low Energy (BLE).

Al igual que el modelo Raspberry Pi 2, también tiene:

- 4 puertos USB.
- 40 pines GPIO.
- Puerto Full HDMI.
- Puerto Ethernet.
- Conector combo compuesto de audio y vídeo de 3,5 mm.
- Interfaz de la cámara (CSI).

- Interfaz de pantalla (DSI).
- Ranura para tarjetas microSD (ahora push-pull en lugar de push-push).
- Núcleo de gráficos Video Core IV 3D.
- Dimensiones de placa de 8.5 por 5.3 cm.
- La Raspberry Pi 3 tiene un factor de forma idéntica a la anterior Pi 2 (1 y Pi Modelo B +) y tiene una compatibilidad completa con Frambuesa Pi 1 y 2.

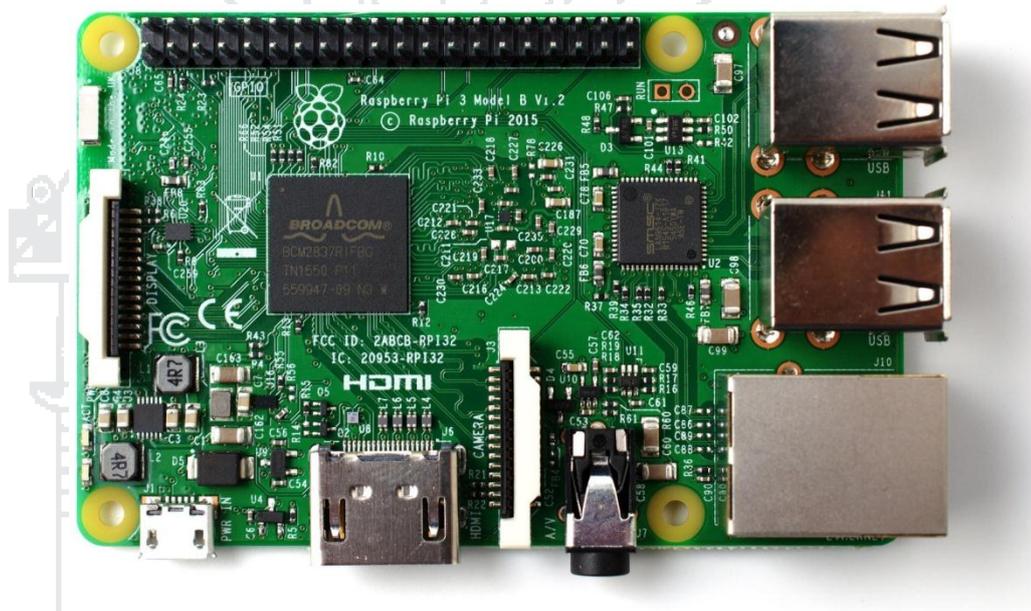


Figura 27: Raspberry PI 3 B+.

Fuente: Warner, T. (2014).

¿Porque Raspberry PI y no otra plataforma de desarrollo como Arduino?

Si bien arduino es una plataforma de desarrollo práctica, ofrece todo un entorno de desarrollo y cuenta con shields que le permiten conectarse a una red ethernet y hasta a una red GSM, y cuentan con conversores ADC incorporados, tiene varios puntos débiles que son su espacio de memoria ROM y RAM, que se llenan fácilmente con las librerías que hay para Shield y código propiamente dicho, además solo pueden correr un script lo que nos obligaría a trabajar mucho más en el código fuente, y solo hay contados modelos que cuentan con un ADC diferencial y con una impedancia de entrada elevada, por lo que, se tendría que buscar otro módulo ADC necesariamente.

Todo esto haría que el número de componentes aumentara lo que acrecentaría su costo, y no todos los shields son fáciles de encontrar en el mercado local, por tanto la Raspberry pi representa la mejor opción en hardware, y costos de implementación.

3.9.6 Memoria micro SD

La memoria micro SD actuara como un disco duro, es donde se cargara el sistema operativo con el que trabajara la Raspberry PI 3 B+ pues esta no cuenta con una memoria de programa, el fabricante recomienda, para que la Raspberry PI B+ trabaje de manera correcta, que la memoria cuente con ciertas características que se listan a continuación:

- Capacidad de memoria: min 8GB, recomendable 32GB, no se especifica límite.
- Categoría: 10 recomendado.
- Tipo: micro SDHC o micro SDXC.

La memoria micro SD que elegimos tiene las siguientes características:

- Capacidad: 32GB.
- Tipo Micro SDHC de categoría 10, que indica una transferencia de datos mínima de 10mbps.



Figura 28: Memoria micro SDHC cat. 10 32 GB.

Elaboración: Propia.

3.10 PROCEDIMIENTO DE DISEÑO, E IMPLEMENTACION

El procedimiento para la implementación consta de los siguientes pasos:

- Instalación del sistema operativo en la memoria micro SD.
- Conexión de los componentes.
- Agregar RTC a la Raspberry PI 3B+.
- Realización del código fuente.
- Configurar modem USB GSM/3G.
- Configurar VPN.
- Realizar script de envío de datos por VPN.
- Pruebas.

Necesitaremos algunas herramientas de software que nos ayudaran a configurar la Raspberry PI 3 B+

3.10.1 Instalación del sistema operativo

Existen dos procedimientos para la instalación del sistema operativo de en el micro SD, pero usaremos un método que nos permitirá instalar el sistema operativo directamente en el micro SD desde una computadora cualquiera:

Como primer paso debemos elegir el sistema operativo que se usara, la Raspberry PI 3 B+ puede soportar varios sistemas operativos que se listan a continuación:

- Raspbian en su distribución Jessie.
- Ubuntu mate.
- Snappy Ubuntu core.
- Windows 10 IOT core.
- OSMC.

- Librelec.
- Pinet.
- RIC OS

Par este propósito se eligió usar la última distribución de Raspbian la Raspbian Jessy, existe una distribución anterior que es la Raspbian Whezzy pero este sistema operativo ya no es compatible con el nuevo procesador que lleva la Raspberry PI 3 B+, otro motivo por el cual elegí este sistema operativo es porque es el más extendido entre los desarrolladores.

- Entonces descargamos la imagen ISO del sistema operativo directamente de la página oficial de Raspberry PI (https://downloads.raspberrypi.org/raspbian_latest), descomprimos el archivo ZIP y guardamos en una carpeta cualquiera.
- Luego necesitamos un programa llamado **Win32DiskImager** que nos permite desglosar la imagen ISO del sistema operativo y grabarla en la memoria SD, este programa lo podemos encontrar fácilmente en el internet y existen muchos sitios web de donde podremos descargarla.
- Usando un adaptador para la memoria Micro SD la conectamos a la computadora.
- Instalamos el programa y al abrirlo nos aparecerá la siguiente ventana, seleccionamos la letra de la memoria que le corresponda a la memoria micro SD y al hacer click en el símbolo de archivo podremos dirigirnos hasta la carpeta donde se encuentra la imagen ISO del sistema operativo. Y le damos click al icono “Write”.

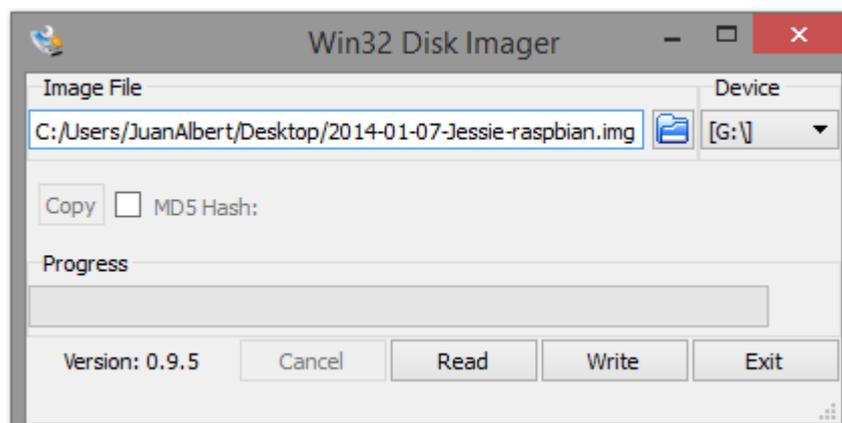


Figura 29: Ventana de Win32DiskImager.

Elaboración: Propia.

- Tardará unos minutos en grabar el sistema operativo en la memoria micro SD y cuando haya finalizado la grabación nos mostrara la siguiente imagen.

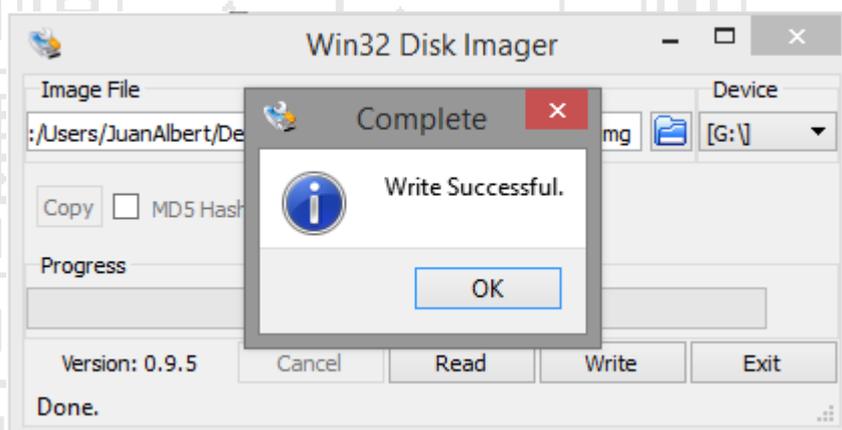


Figura 30: Grabación finalizada.

Elaboración: Propia.

- Insertamos la memoria micro SD en el slot de la Raspberry PI 3B+ , alimentamos la Raspberry PI, esperamos a que el sistema inicie y a partir de este punto, si contamos con un teclado, un raton y un monitor con puerto HDMI podremos hacer todas las configuraciones directamente en la Raspberry PI, pero en caso no contemos con esos aditamentos podemos acceder en modo consola a la Raspberry PI 3+ a través de SSH (secure shell)

con un pequeño programa llamado Putty que se puede descargar del siguiente enlace (www.putty.org), al abrirlo nos aparecerá la siguiente ventana.

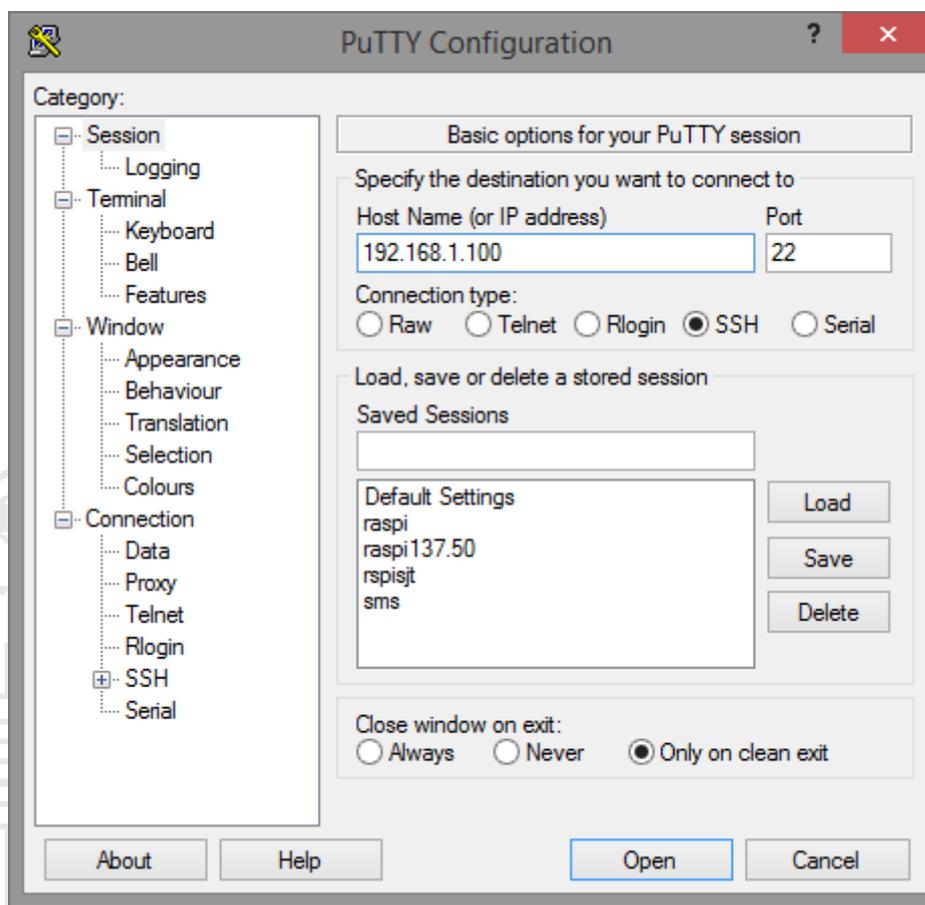


Figura 31: Interfaz de Putty.

Elaboración: Propia.

- Conectamos la Raspberry PI al router con un cable ethernet para que el asigne una dirección IP, pues por defecto la raspberry pi tiene el DHCP activado, pero previamente debemos tener instalada una aplicación que nos permita escanear las direcciones IP para este caso usaremos la aplicación llamada Advanced IP scanner que se puede descargar gratuitamente desde el siguiente enlace (www.advanced-ip-scanner.com), que tras el escaneo nos mostrara la **Figura 32**.

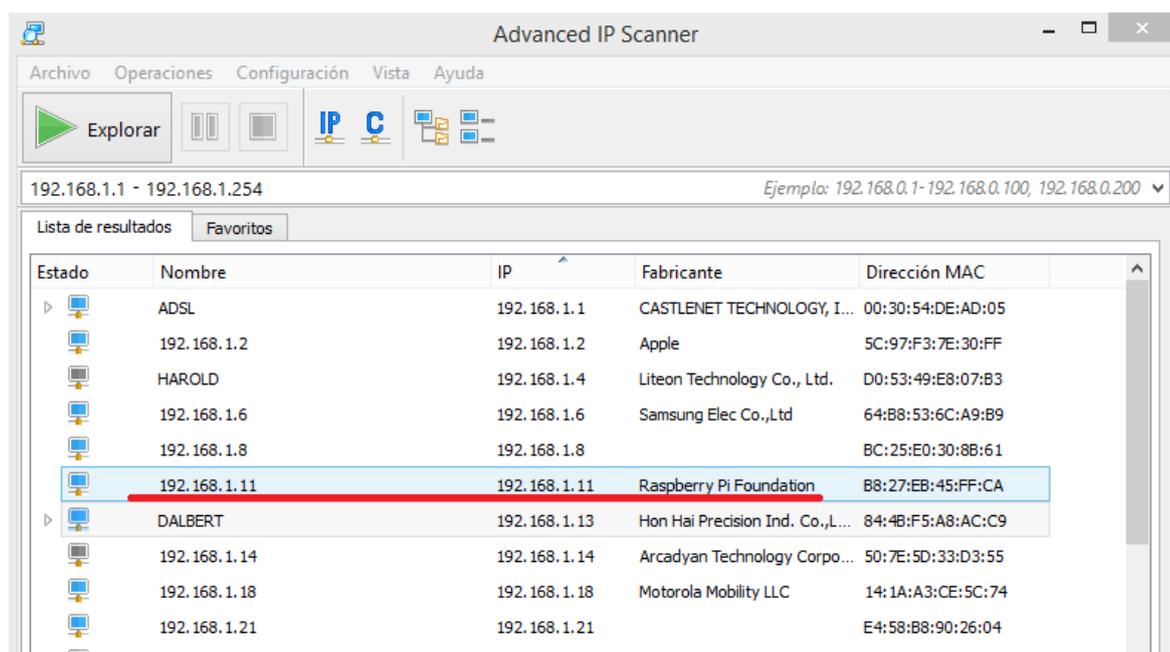


Figura 32: Escaneando de Raspberry PI 3 B+.

Elaboración: Propia.

- El siguiente paso, para ya no necesitar nuevamente el escaneado le daremos una IP fija a nuestra Raspberry PI 3B+.
- Nos dirigimos a la ventana de del software Putty y escribiremos la IP encontrada en este caso 192.168.1.11 y le damos a “OPEN”.

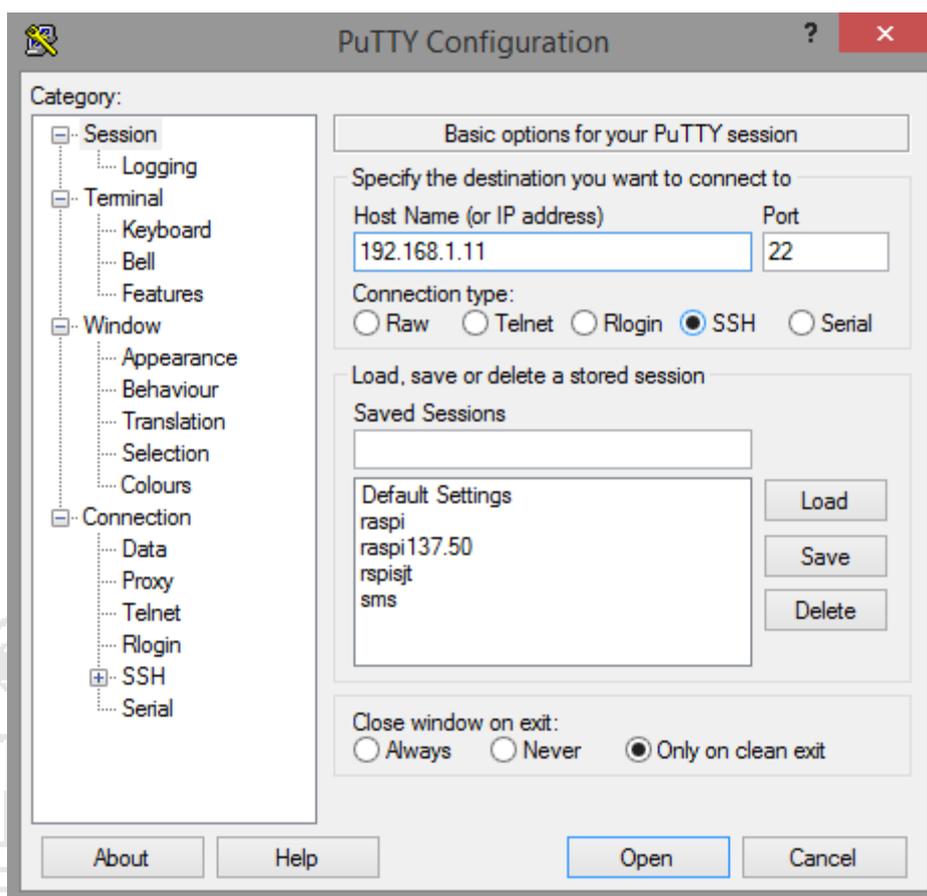


Figura 33: Accediendo a la Raspberry PI.

Elaboración: Propia.

- Y nos saldrá la siguiente pantalla que muestra la **Figura 34**, que nos pide un usuario y contraseña que por defecto es usuario = PI , contraseña: raspberry.

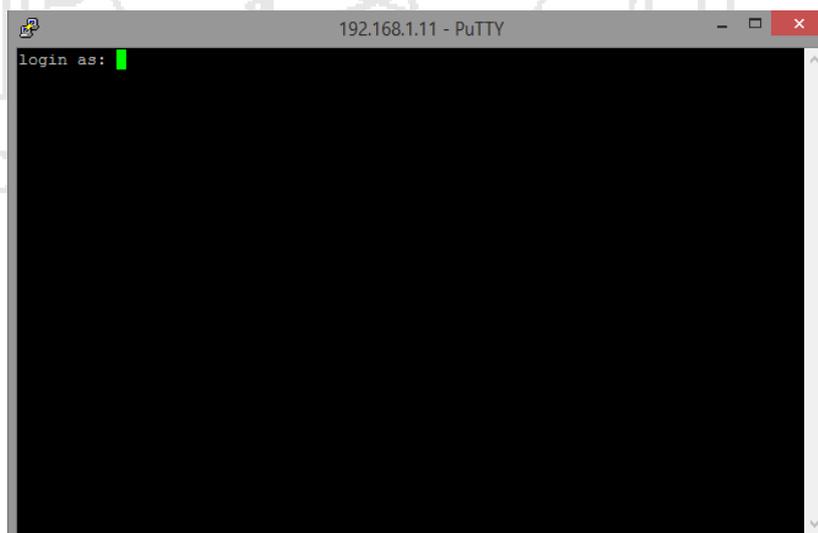


Figura 34: Consola de Raspberry pi en Putty.

Elaboración: Propia.

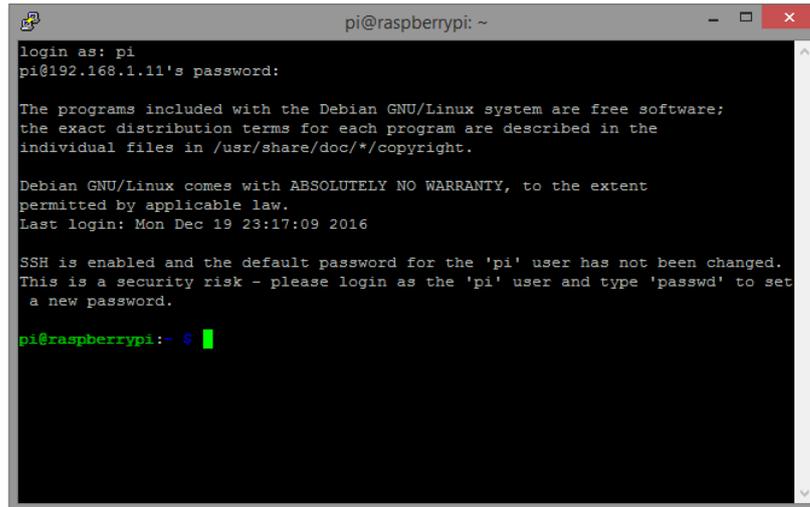


Figura 35: Ventanas de Putty una vez dentro de Raspberry PI.

Elaboración: Propia.

- Luego escribimos los siguientes comandos:
- sudo nano /etc/network/interfaces
- Y obtenemos lo siguiente:

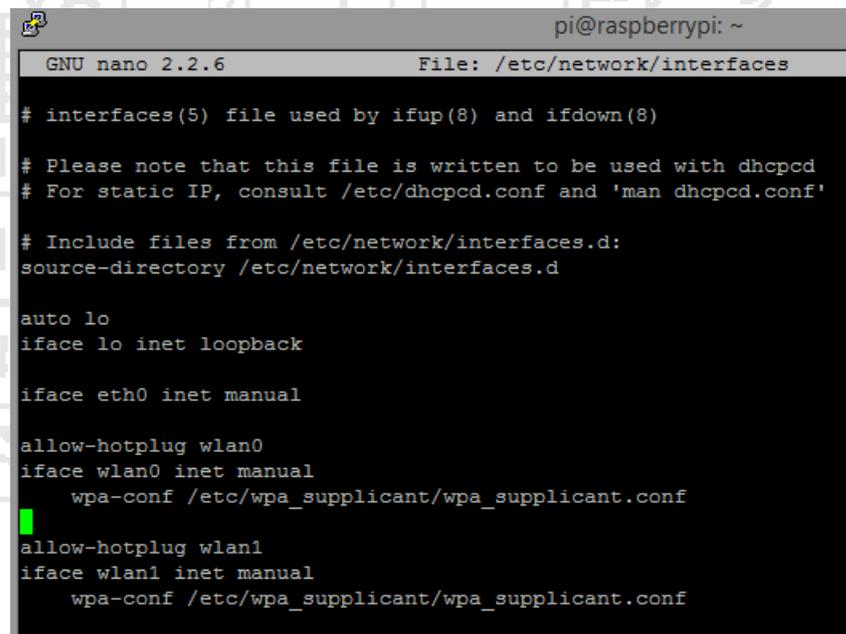


Figura 36: Archivo de configuración de IP.

Elaboración: Propia.

- Debemos cambiar todo el texto a partir de “auto lo”

Por las siguientes líneas:

auto lo

```
iface lo inet loopback

iface eth0 inet static

address 192.168.1.15 // IP que se asignara al puerto ethernet de la raspberry

pi

netmask 255.255.255.0

network 192.168.1.0

broadcast 192.168.1.255

gateway 192.168.1.254

allow-hotplug wlan0

auto wlan0

iface wlan0 inet static

address 192.168.1.100 // dirección IP que se le asignara al controlador WIFI

netmask 255.255.255.0

network 192.168.1.0

broadcast 192.168.1.255

gateway 192.168.1.254

wpa-ssid "nombre_de_red"

wpa-psk "contraseña"
```

Reiniciamos la Raspberry para que los cambios se realicen con el siguiente comando (Pipe, 2015).

- Sudo reboot
- A partir de este momento podemos conectarnos a la Raspberry PI inalámbricamente, abrimos Putty y entramos por cualquiera de las nuevas direcciones IP nuevas que se asignó en el texto anterior, una vez dentro de

Raspberry PI es necesario comprobar que tiene conexión a internet y para ello usaremos el siguiente comando

- `wget www.google.com`

Si es que hay conexión a internet nos descargara un pequeño archivo html y sino deberemos revisar todo y asegurarnos que hay conexión a internet

- El siguiente paso es actualizar el sistema operativo con los siguientes comandos, y se dará por concluida la parte de instalación del sistema operativo ahora podremos continuar con las configuraciones.

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

3.10.1.1 Instalación y configuración de servidor FTP (vsftpd)

Vsftp (very secure ftp daemon), es un pequeño pero seguro servidor ftp, que corre en segundo plano en nuestra Raspberry, es una herramienta muy útil que permitirá transferir archivos entre la Raspberry PI 3 B+ y otras computadoras que estén conectadas a una misma red, para instalarlo es necesario estar conectado a internet pues al correr el comando automáticamente la Raspberry se conecta a los sitios web donde están guardados estos archivos, para instalarlo se sigue el siguiente procedimiento (Perez, 2013).

Ejecutamos el siguiente comando

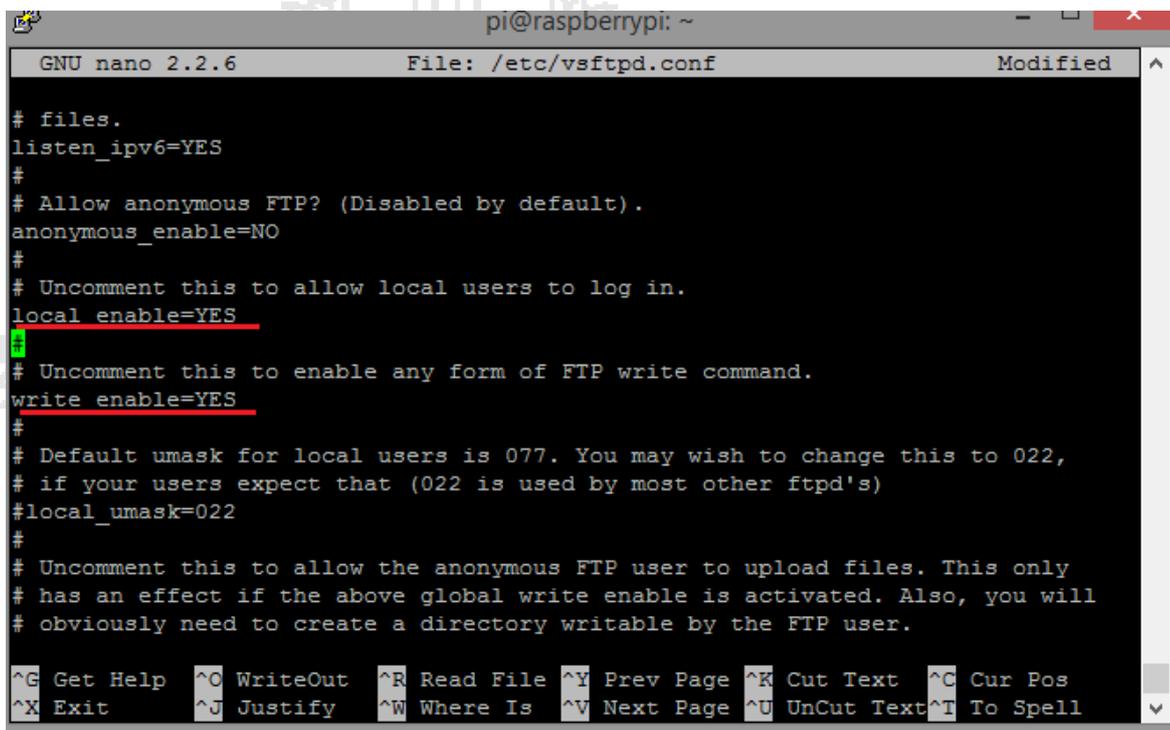
- `sudo apt-get install vsftpd`

Ahora debemos habilitarlo de manera que permita el acceso de otra computadora mediante un cliente FTP que podría ser el FileZilla o cualquier otro. Para habilitarlo abrimos el archivo de configuración `vsftpd.conf` con el siguiente comando:

- `sudo nano /etc/vsftpd.conf`

Y ubicamos y des comentamos las líneas como muestra la **Figura 37.**:

- `local_enable=YES`
- `write_enable=YES`



```

pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/vsftpd.conf Modified
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Figura 37: Archivo vsftpd.conf.

Elaboración: Propia.

Reiniciamos el servicio vsftpd con el siguiente comando y tendremos nuestro servidor ftp funcionando.

- `sudo service vsftpd restart`

3.10.2 Conexión de componentes

Una vez tenemos la Raspberry pi con el sistema operativo instalado en la memoria micro SD armamos el siguiente esquema

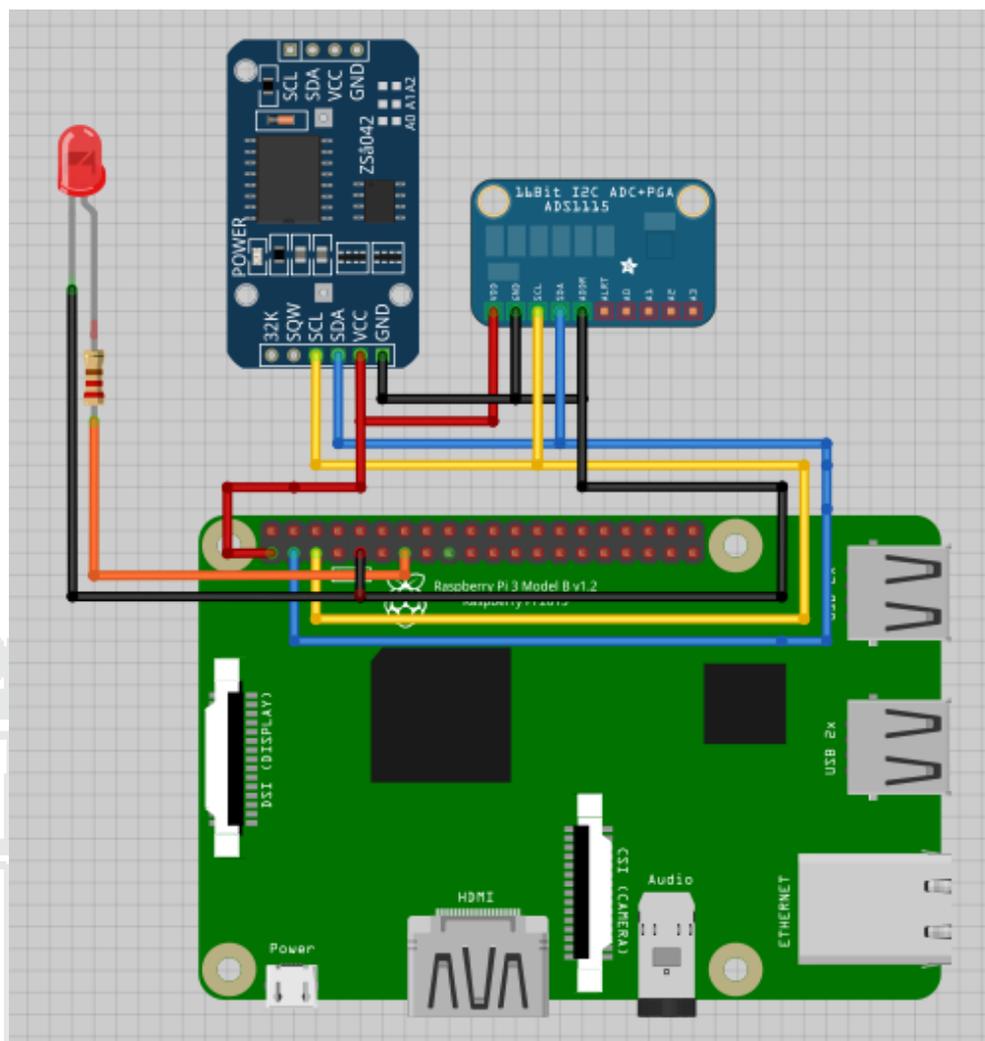


Figura 38: Esquema de conexión de módulos.

Elaboración: Propia.

3.10.3 Agregar RTC a Raspberry PI 3 B+

Lo primero que se debe hacer para agregar un reloj en tiempo real y que el sistema tome la hora de este reloj es habilitar el bus I2C de la Raspberry, esto se logra accediendo al sistema mediante Putty mediante ssh y en la consola entramos al modo configuración con los siguientes comandos

- Sudo raspi-config

Nos aparecerá la siguiente ventana

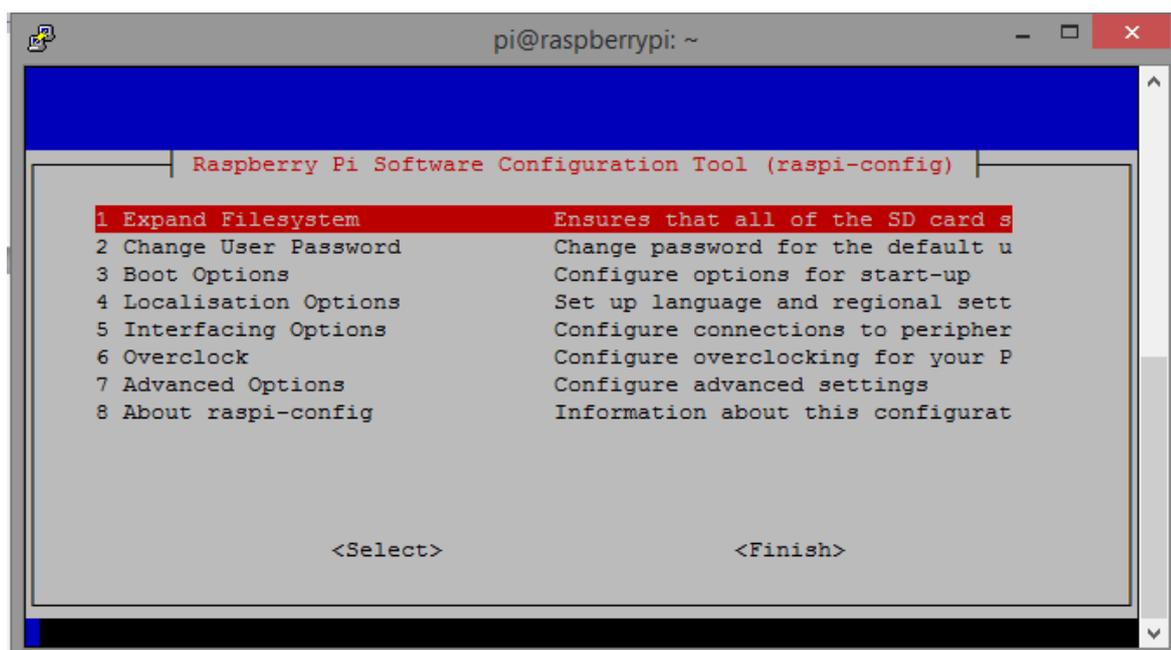


Figura 39: Ventana de configuración de Raspberry.

Elaboración: Propia.

Navegamos por las opciones con las teclas de dirección y damos enter en “Interfacing options” y ubicaremos la opción “I2C” damos enter y nos saldrá una ventana donde deberemos seleccionar “Yes”



Figura 40: Ventana de habilitación de del Bus I2C.

Elaboración: Propia.

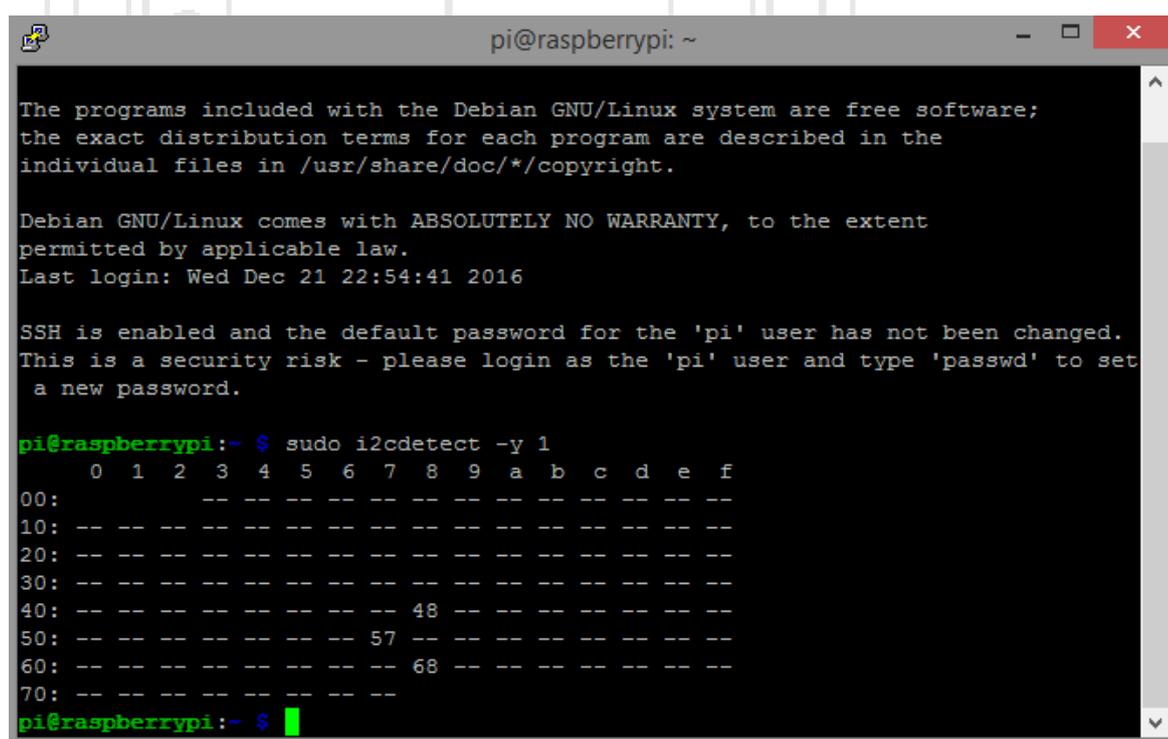
Reiniciamos el sistema para que los cambios se efectúen. Volvemos a entrar en el sistema y corremos los siguientes comandos que instalan herramientas de manejo del bus I2C.

- `sudo apt-get install python-smbus`
- `sudo apt-get install i2c-tools`

Luego para verificar que el bus I2C está activo ejecutamos el siguiente comando que nos indicara todos los dispositivos conectados al bus mencionado. (Matt, 2015)

- `sudo i2cdetect -y 1`

Tras ejecutar este comando nos mostrara la siguiente imagen:



```

pi@raspberrypi: ~
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 21 22:54:41 2016

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~$ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  48  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  57  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~$

```

Figura 41: Componentes conectados al bus I2C.

Elaboración: Propia.

Vemos que aparecen tres números 48, 57, 68, de los cuales 48 corresponde al módulo ADC ADS1115 y el 68 corresponde al módulo RTC DS3231, y 57 no tiene

ninguna utilidad. Ahora que tenemos el bus habilitado debemos establecerlo como fuente de hora por defecto, para esto hacemos el siguiente procedimiento.

Escribimos el siguiente comando

- `sudo nano /boot/config.txt`

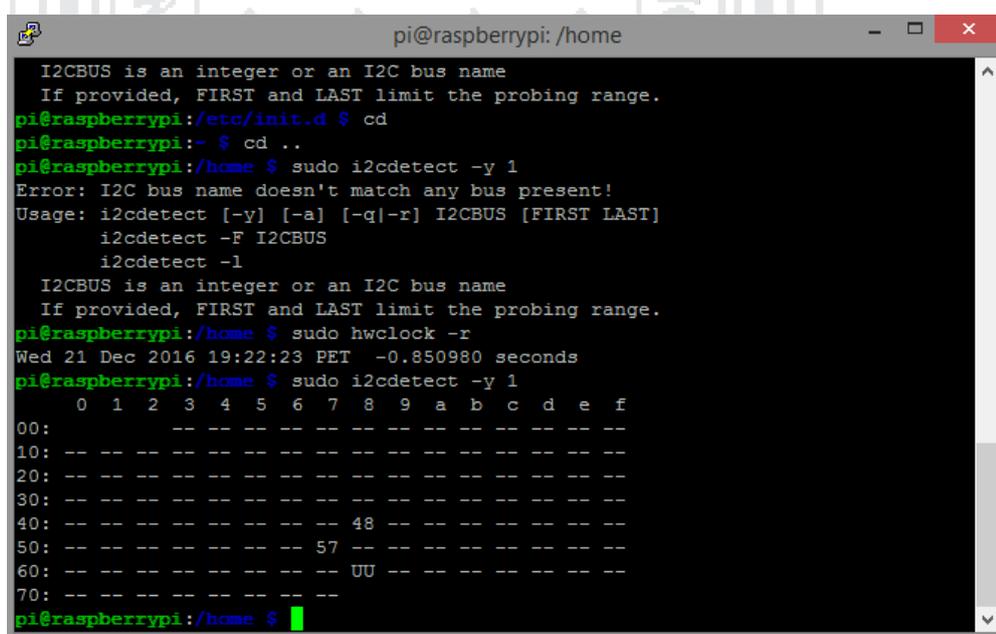
Y agregamos la siguiente línea de código

- `dtoverlay=i2c-rtc,ds3231`
- `dtparam=i2c_arm=on`
- `sudo reboot`

Para comprobar que el RTC ya está instalado en Kernel comprobamos con el comando

- `sudo i2cdetect -y 1`

La **Figura 42** Muestra “UU” en donde se encontraba la dirección del RTC, que es un indicativo de que el bus en esa dirección está siendo usado por el sistema.



```

pi@raspberrypi: /home
I2CBUS is an integer or an I2C bus name
If provided, FIRST and LAST limit the probing range.
pi@raspberrypi:/etc/init.d $ cd
pi@raspberrypi:~ $ cd ..
pi@raspberrypi:/home $ sudo i2cdetect -y 1
Error: I2C bus name doesn't match any bus present!
Usage: i2cdetect [-y] [-a] [-q|-r] I2CBUS [FIRST LAST]
i2cdetect -F I2CBUS
i2cdetect -l
I2CBUS is an integer or an I2C bus name
If provided, FIRST and LAST limit the probing range.
pi@raspberrypi:/home $ sudo hwclock -r
Wed 21 Dec 2016 19:22:23 PET -0.850980 seconds
pi@raspberrypi:/home $ sudo i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  48  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  57  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:/home $

```

Figura 42: RTC instalado en el Kernel.

Elaboración: Propia.

El siguiente paso es inhabilitar el servidor NTP para ello ejecutamos los siguientes comandos.

- Sudo update-rc -f ntp remove
- apt-get purge fake-hwclock

Para comprobarlo podemos fijar una hora cualquiera y comprobamos que la hora que tiene la Raspberry es del RTC. Con los siguientes comandos.

- sudo date 08020503
- sudo hwclock -w
- sudo reboot
- date

Si la hora y fecha corresponde con la ingresada manualmente entonces todo está correcto. (Jan, 2016)

3.10.4 Realización del código fuente

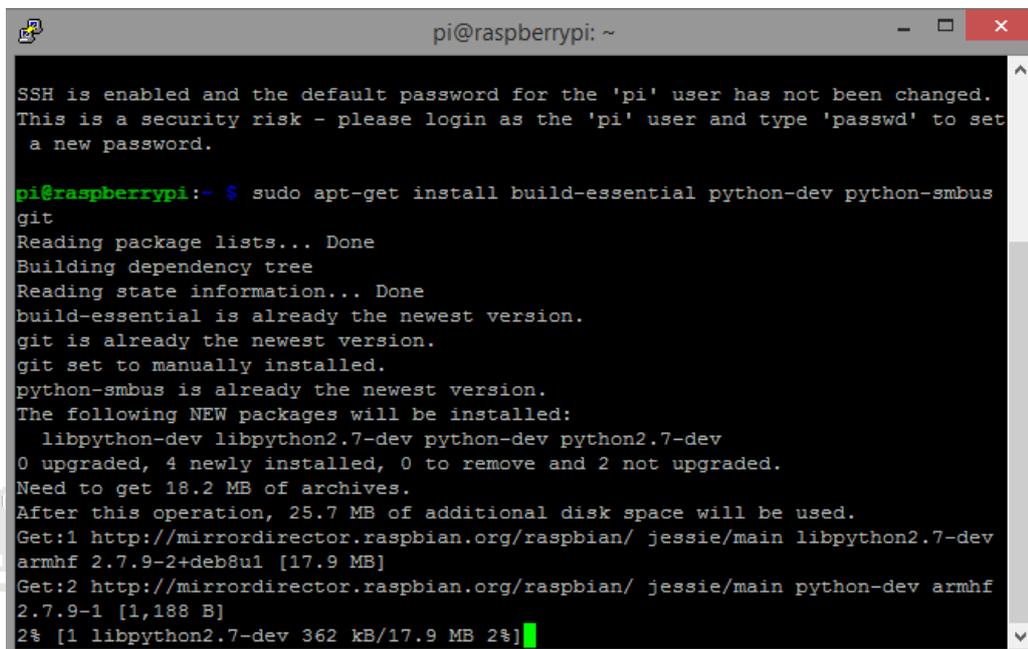
Para la realización del programa se usó Python 3 que viene por defecto en el sistema operativo de la Raspberry PI, el programa consta de una sola rutina que maneja solo ADS1115 puesto que el RTC y los demás componentes es manejado directamente por el sistema operativo, de hecho es una ventaja que tiene Raspberry. Para la realización del código es necesario realizar los siguientes pasos.

3.10.4.1 Instalación de paquetes de python

La Raspberry "PI 3 B+" es compatible con las versiones de Python 2.7 y Python 3, que ya vienen instaladas por defecto en sistema operativo.

- Verificar e instalar paquetes de Python, abrimos Putty y ejecutamos los siguientes comandos.

- `sudo apt-get install build-essential python-dev python-smbus git`



```

pi@raspberrypi: ~
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

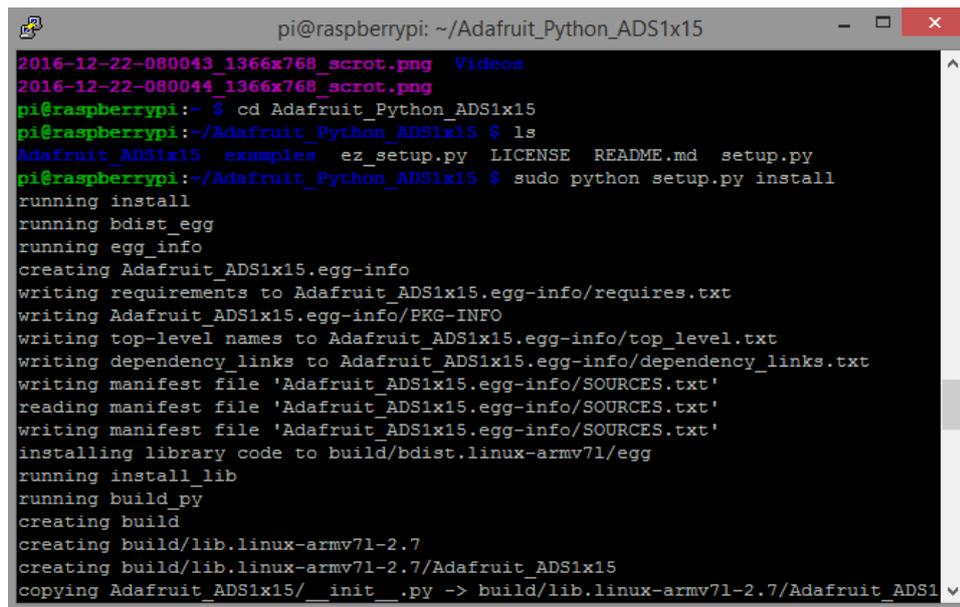
pi@raspberrypi:~$ sudo apt-get install build-essential python-dev python-smbus
git
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version.
git is already the newest version.
git set to manually installed.
python-smbus is already the newest version.
The following NEW packages will be installed:
  libpython-dev libpython2.7-dev python-dev python2.7-dev
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.
Need to get 18.2 MB of archives.
After this operation, 25.7 MB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main libpython2.7-dev
armhf 2.7.9-2+deb8u1 [17.9 MB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main python-dev armhf
2.7.9-1 [1,188 B]
2% [1 libpython2.7-dev 362 kB/17.9 MB 2%]

```

Figura 43: Instalando paquetes de Python.

Elaboración: Propia.

- Librería para el ADS1115 en Python, que se puede descargar e instalar con los siguientes comandos.
 - `cd ~`
 - `git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git`
 - `cd Adafruit_Python_ADS1x15`
 - `sudo python setup.py install`



```

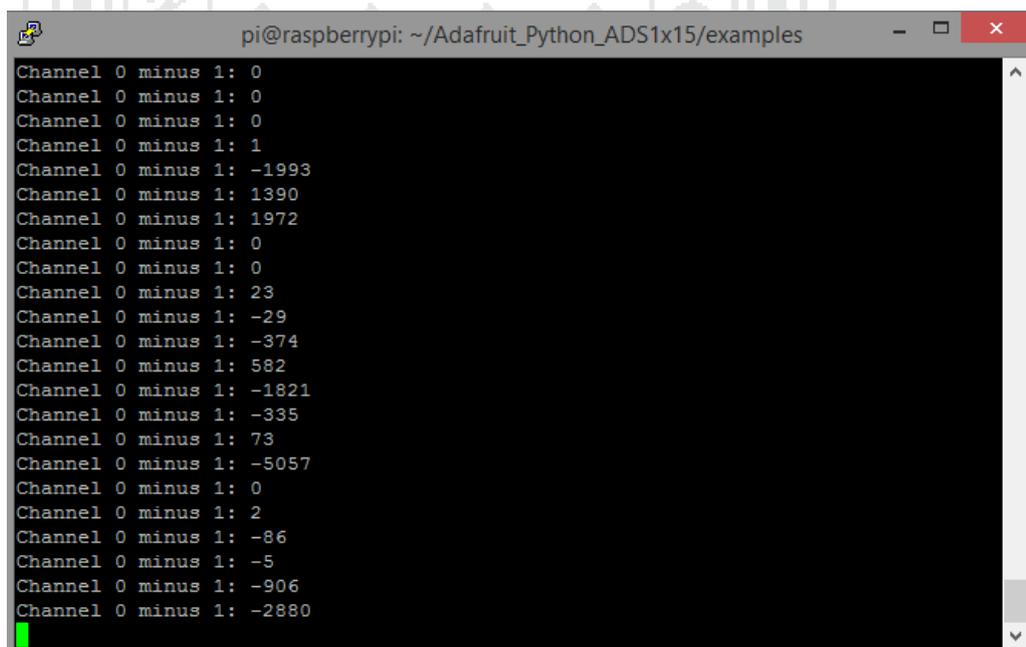
pi@raspberrypi: ~/Adafruit_Python_ADS1x15
2016-12-22-080043 1366x768_scrot.png Videos
2016-12-22-080044 1366x768_scrot.png
pi@raspberrypi:~ $ cd Adafruit_Python_ADS1x15
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $ ls
Adafruit_ADS1x15  examples  ez_setup.py  LICENSE  README.md  setup.py
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $ sudo python setup.py install
running install
running bdist_egg
running egg_info
creating Adafruit_ADS1x15.egg-info
writing requirements to Adafruit_ADS1x15.egg-info/requires.txt
writing Adafruit_ADS1x15.egg-info/PKG-INFO
writing top-level names to Adafruit_ADS1x15.egg-info/top_level.txt
writing dependency links to Adafruit_ADS1x15.egg-info/dependency_links.txt
writing manifest file 'Adafruit_ADS1x15.egg-info/SOURCES.txt'
reading manifest file 'Adafruit_ADS1x15.egg-info/SOURCES.txt'
writing manifest file 'Adafruit_ADS1x15.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-armv7l/egg
running install_lib
running build_py
creating build
creating build/lib.linux-armv7l-2.7
creating build/lib.linux-armv7l-2.7/Adafruit_ADS1x15
copying Adafruit_ADS1x15/_init_.py -> build/lib.linux-armv7l-2.7/Adafruit_ADS1

```

Figura 44: Instalando librerías para ADS1115.

Elaboración: Propia.

- Para comprobar que la librería está correctamente instalada comprobamos con uno de los ejemplos que trae por defecto.
 - cd Adafruit_Python_ADS1x15
 - cd examples
 - sudo python differential.py



```

pi@raspberrypi: ~/Adafruit_Python_ADS1x15/examples
Channel 0 minus 1: 0
Channel 0 minus 1: 0
Channel 0 minus 1: 0
Channel 0 minus 1: 1
Channel 0 minus 1: -1993
Channel 0 minus 1: 1390
Channel 0 minus 1: 1972
Channel 0 minus 1: 0
Channel 0 minus 1: 0
Channel 0 minus 1: 23
Channel 0 minus 1: -29
Channel 0 minus 1: -374
Channel 0 minus 1: 582
Channel 0 minus 1: -1821
Channel 0 minus 1: -335
Channel 0 minus 1: 73
Channel 0 minus 1: -5057
Channel 0 minus 1: 0
Channel 0 minus 1: 2
Channel 0 minus 1: -86
Channel 0 minus 1: -5
Channel 0 minus 1: -906
Channel 0 minus 1: -2880

```

Figura 45: Comprobando que el modulo funciona correctamente.

Elaboración: Propia.

- Ahora creamos una carpeta donde se alojarán todos los scripts referentes al prototipo
 - mkdir scripts
- Y creamos un archivo con extensión “py” donde describiremos el código fuente.
 - cd scripts
 - nano lee_ADC.py

3.10.4.2 Diagrama de flujo del programa

Luego realizamos el diagrama de flujo del programa para entender mejor lo que se busca realizar, la **Figura 46** muestra el diagrama de flujo del programa escrito en Python, inicialmente verifica si ya existe la dirección del archivo, si existe continua con la rutina y si no existe crea las carpetas, el siguiente proceso primer proceso toma la hora actual y le suma el periodo de esa manera el programa sabe a qué hora tomara la siguiente muestra, la primera condicional se asegura de que el muestreo siempre inicie en el segundo “0” para tener un mejor orden, cuando esta condición se cumple recién entra al bucle principal donde se encuentra con la condicional que indica que la hora actual es o no igual a la hora siguiente que se estableció al inicio, que es la hora presente anterior más el periodo o tiempo entre muestreo y muestreo. Los siguientes procesos son el muestro del voltaje y multiplicación por un factor para convertirlo a un valor en voltaje y apertura de carpetas donde se guardan los valores medidos.

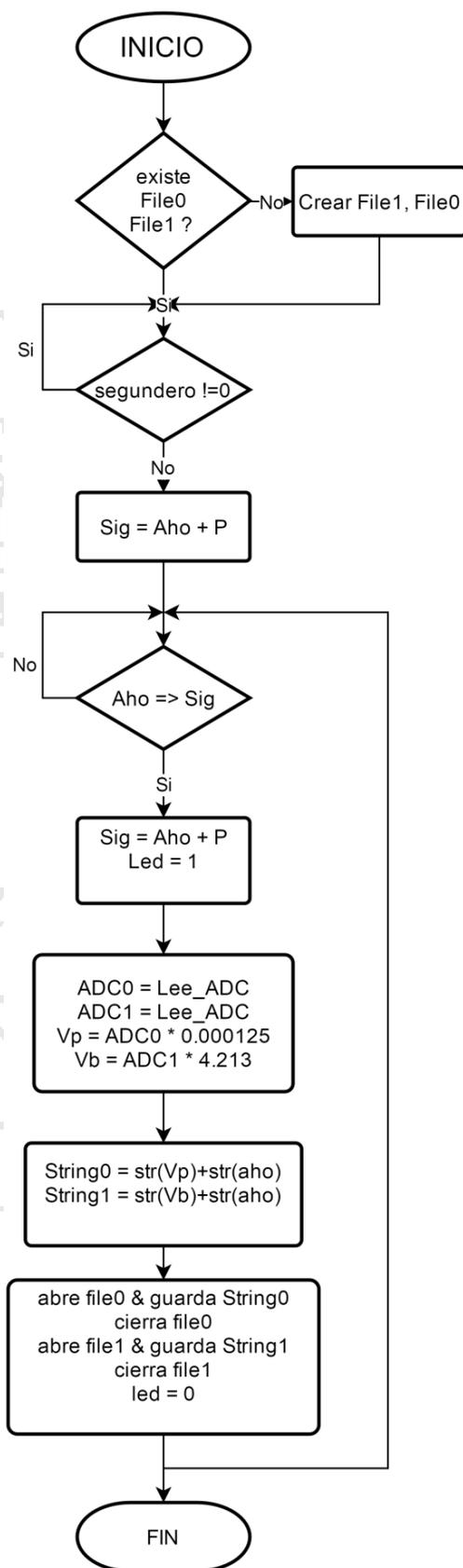


Figura 46: Diagrama de flujo para el programa.

Elaboración: Propia.

3.10.4.3 Inicio automático del programa

Para que el programa inicie al arrancar el sistema necesitamos crear un script de inicio en el directorio `/etc/init.d`, e script es una lista de comandos que se ejecutaran después de que todos los servicios (demonios) del sistema hayan arrancado (Gorkau, 2013), para ello nos dirigimos al directorio mencionado y creamos un nuevo archivo:

- `cd /etc/init.d`
- `sudo nano autoinicio_ADC`

Una vez abierto el archivo escribimos las siguientes líneas de comandos:

```
#!/bin/sh
# /etc/init.d/volt_start
### BEGIN INIT INFO
# Provides:      autoinicio_ADC
# Required-Start:  $all
# Required-Stop:  $remote_fs $syslog
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Script de ejemplo de arranque automático
# Description:    Script para muestrear potencial espontaneo
### END INIT INFO
# Dependiendo de los parámetros que se le pasen al programa se usa una opción u otra
case "$1" in
start)
    sleep 10
    echo "Arrancando autoinicio_ADC "
    # Aquí hay que poner el programa que quieras arrancar automáticamente
    /usr/bin/python /home/pi/scripts/ lee_ADC.py
    ;;
stop)
    echo "Deteniendo autoinicio_ADC "
    killall /usr/bin/python /home/pi/scripts/ lee_ADC.py
    ;;
```

*)

```
echo "Modo de uso: /etc/init.d/ autoinicio_ADC {start|stop}"
exit 1
;;
esac
exit 0
#*****
```

Guardamos y cerramos, para que pueda ser ejecutado es necesario dotarle de permisos de súper usuario esto se logra con el siguiente comando:

- sudo chmod 755 /etc/init.d/autoinicio_ADC

Para comprobar el correcto funcionamiento de auto arranque ejecutamos el siguiente comando

- sudo /etc/init.d/ **autoinicio_ADC** start

Debería de arrancar el programa hecho en Python, finalmente activamos el arranque automático con el comando:

- sudo update-rc.d **autoinicio_ADC** defaultssudo reboot

3.10.5 Configuración de modem USB GSM/3G

El modem USB que se usa en este prototipo es el MF180 de la marca ZTE, cabe mencionar que en la versión anterior de Raspbian (wheezy), estos módems eran reconocidos como unidad de almacenamiento por tanto se debía seguir un procedimiento para cambiar el modo en que eran reconocidos que es la de un modem USB (Vasquez, 2013), esto ya no sucede en la nueva versión de Raspbian (jessie), lo cual podemos comprobar con los siguientes comandos.

- lsusb

Este comando lista los dispositivos conectados a los puertos USB de la Raspberry Pi 3 y cuyo resultado no muestra la **Figura 47**.

```

pi@raspberrypi: ~
[43844.750751] usb 1-1.3: New USB device strings: Mfr=3, Product=2, SerialNumber=4
[43844.750760] usb 1-1.3: Product: ZTE WCDMA Technologies MSM
[43844.750767] usb 1-1.3: Manufacturer: ZTE, Incorporated
[43844.750774] usb 1-1.3: SerialNumber: MF1800ZTED010000
[43844.763710] option 1-1.3:1.0: GSM modem (1-port) converter detected
[43844.764610] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[43844.765180] option 1-1.3:1.1: GSM modem (1-port) converter detected
[43844.765669] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[43844.766222] usb-storage 1-1.3:1.2: USB Mass Storage device detected
[43844.766538] scsi host3: usb-storage 1-1.3:1.2
[43844.767368] option 1-1.3:1.3: GSM modem (1-port) converter detected
[43844.769151] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[43845.767972] scsi 3:0:0:0: Direct-Access    ZTE      MMC Storage    2.31 PQ
: 0 ANSI: 2
[43845.768855] sd 3:0:0:0: Attached scsi generic sg0 type 0
[43845.773593] sd 3:0:0:0: [sda] Attached SCSI removable disk
pi@raspberrypi:~$ lsusb
Bus 001 Device 010: ID 19d2:0031 ZTE WCDMA Technologies MSM MF110/MF627/MF636
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~$
  
```

Figura 47: Modem reconocido correctamente.

Elaboración: Propia.

Luego corremos el siguiente comando de diagnóstico que nos indicara como está siendo reconocido nuestro modem, con el comando:

- dmesg

```

pi@raspberrypi: ~
[43842.438930] usb 1-1.3: SerialNumber: MF1800ZTED010000
[43842.441604] usb-storage 1-1.3:1.0: USB Mass Storage device detected
[43842.441798] scsi host2: usb-storage 1-1.3:1.0
[43844.405498] usb 1-1.3: USB disconnect, device number 9
[43844.645954] usb 1-1.3: new high-speed USB device number 10 using dwc_otg
[43844.750717] usb 1-1.3: New USB device found, idVendor=19d2, idProduct=0031
[43844.750751] usb 1-1.3: New USB device strings: Mfr=3, Product=2, SerialNumber=4
[43844.750760] usb 1-1.3: Product: ZTE WCDMA Technologies MSM
[43844.750767] usb 1-1.3: Manufacturer: ZTE, Incorporated
[43844.750774] usb 1-1.3: SerialNumber: MF1800ZTED010000
[43844.763710] option 1-1.3:1.0: GSM modem (1-port) converter detected
[43844.764610] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[43844.765180] option 1-1.3:1.1: GSM modem (1-port) converter detected
[43844.765669] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[43844.766222] usb-storage 1-1.3:1.2: USB Mass Storage device detected
[43844.766538] scsi host3: usb-storage 1-1.3:1.2
[43844.767368] option 1-1.3:1.3: GSM modem (1-port) converter detected
[43844.769151] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[43845.767972] scsi 3:0:0:0: Direct-Access    ZTE      MMC Storage    2.31 PQ
: 0 ANSI: 2
[43845.768855] sd 3:0:0:0: Attached scsi generic sg0 type 0
[43845.773593] sd 3:0:0:0: [sda] Attached SCSI removable disk
pi@raspberrypi:~$
  
```

Figura 48: Modem reconocido como modem USB.

Elaboración: Propia.

En la **Figura 48** vemos que el modem está siendo detectado como un modem GSM, ahora solo queda instalar una aplicación que conecte la Raspberry pi 3 al internet a través del modem, que recupere la conexión automáticamente si se llegara a desconectar por alguna razón, pero previamente necesitamos instalar un protocolo de punto a punto o PPP con el comando siguiente:

- sudo apt-get install ppp

Por último la aplicación debe iniciar al encender la Raspberry PI 3 B+ (Elias, 2015).

Primero creamos un directorio donde descargaremos la aplicación con los siguientes comandos:

- mkdir umtskeeper
- cd umtskeeper

Con el siguiente comando descargaremos la aplicación

- wget <http://mintakaconciencia.net/squares/umtskeeper/src/umtskeeper.tar.gz>

Descomprimos y listamos el contenido

- md5sum umtskeeper.tar.gz
- tar -xzf umtskeeper.tar.gz
- ls

La **Figura 49** nos muestra el contenido de del archivo que acabamos de descargar

```
pi@raspberrypi:~/umtskeeper $ ls
changelog.txt  LICENSE.txt  README.txt  sakis3g  umtskeeper
favicon.ico    README.html  robots.txt  style.css  umtskeeper.conf.sample
```

Figura 49: Contenido de la carpeta umtskeeper.

Elaboración: Propia.

En la **Figura 49** observamos que hay dos objetos escritos en verde, son los scripts que deberemos correr en la Raspberry PI 3 B+, sakis3G es un script que se encarga de gestionar el modem iniciando la conexión a internet a través del modem USB (Vasquez, 2013), y umtskeeper es un script encargado de mantener la conexión del modem al internet en caso de que este llegara a desconectarse por algún motivo. (Elias, 2015). Luego es necesario comprobar que el script sakis3G funciona correctamente con el siguiente comando:

- `sudo ./sakis3g -interactive`

Al ejecutar el comando debería mostrar la siguiente imagen, las imágenes posteriores muestran el procedimiento que se debería seguir para comprobar que la Raspberry PI 3 B+ se conecta a internet

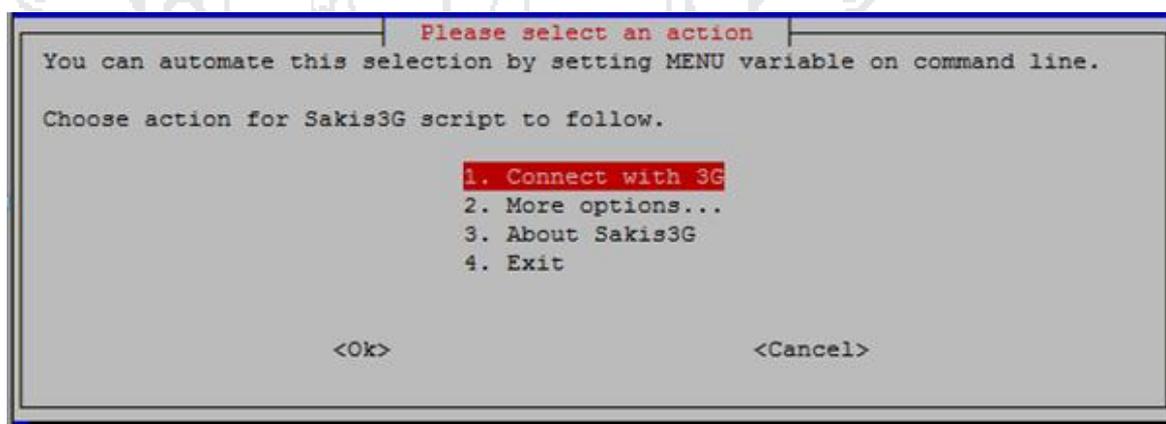


Figura 50: Interfaz de Sakis3G.

Elaboración: Propia.

Elegimos USB device, porque se trata de un dispositivo USB.

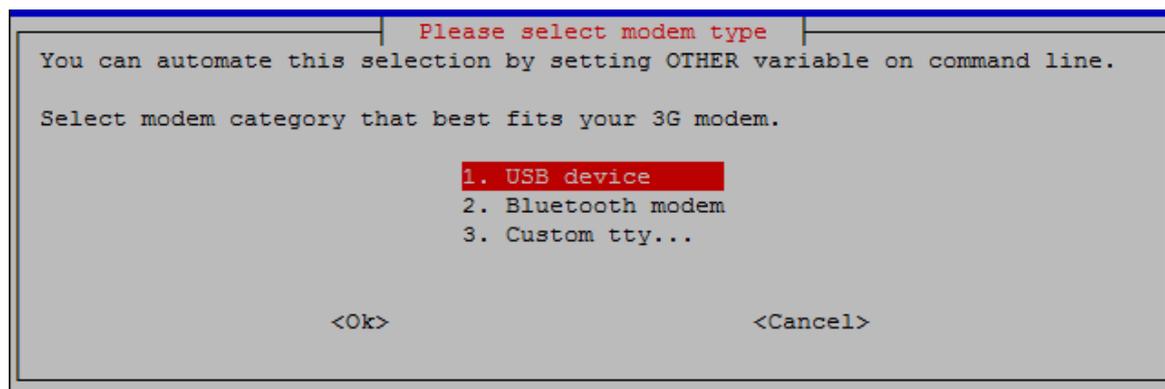


Figura 51: Selección de dispositivo.

Elaboración: Propia.

Elegimos el ZTE WCDMA Technologies MSM, pues es la marca de nuestro modem.

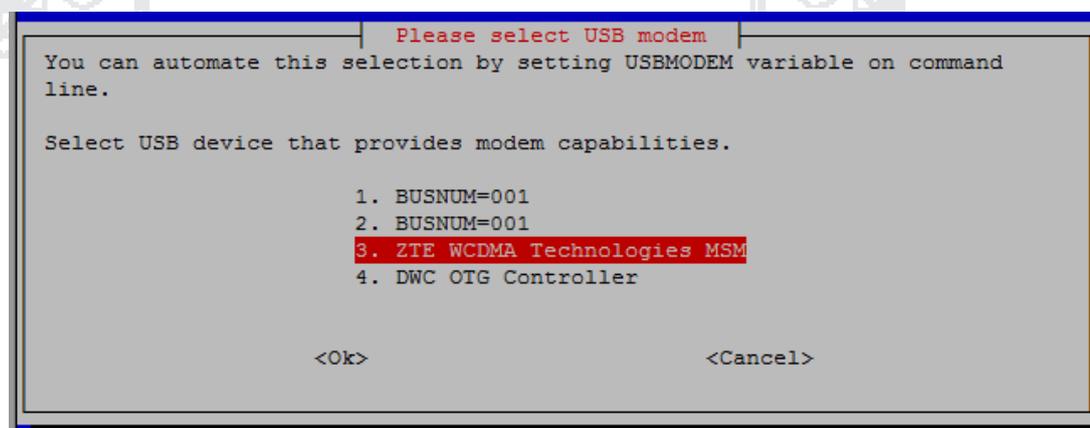


Figura 52: Selección de modem.

Elaboración: Propia.

Elegimos el APN que encuentra por defecto, para cuando esté completamente instalado usaremos otro APN, pero siendo esta solo una prueba usaremos el APN por defecto.

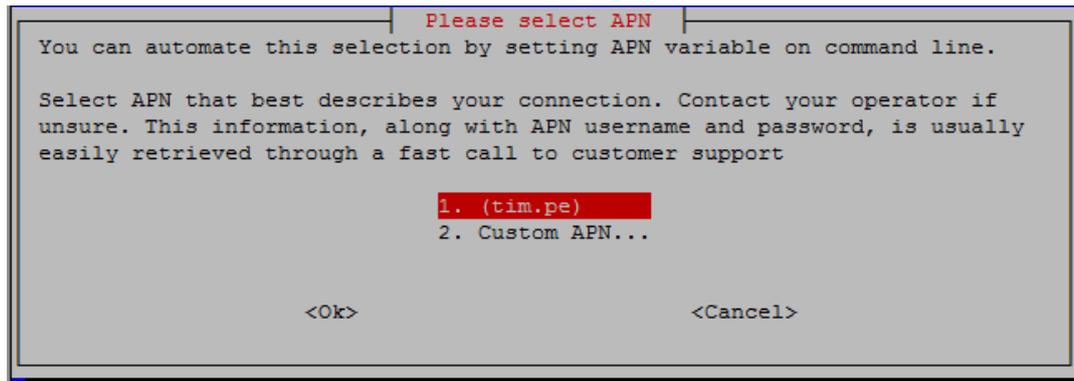


Figura 53: Selección de APN.

Elaboración: Propia.

La **Figura 54** muestra el mensaje de que el modem se ha conectado correctamente y le damos a “OK”.

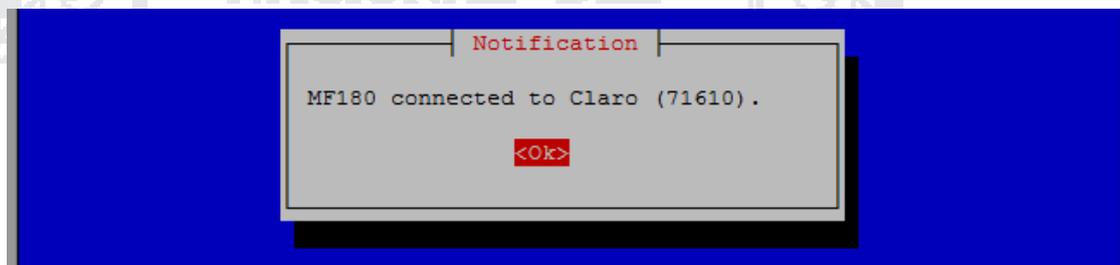


Figura 54: Mensaje de confirmación de Sakis3G.

Elaboración: Propia.

En este punto tenemos el modem conectado pero es susceptible a que se pierda la conexión cuyas posibles causas pueden ser que la señal de la operadora caiga (lo cual es bastante frecuente), que el sistema operativo se reinicie, apagar y encender la Raspberry PI, etc. por tanto necesitamos un script que mantenga la conexión permanente y si se llegara a desconectar por algún motivo, este debería de reconectarse automáticamente. Esta tarea la hace el UMTSKEEPER, que constante mente verifica que la conexión está establecida, y en caso de que se hubiese perdido reinicia el script sakis3G, para su funcionamiento necesita los parámetros necesarios para que sakis3G pueda conectarse, siendo estos parámetros:

- ID del modem USB: 19d2:0031
- APN de operadora: claro.pe

- APN usuario: claro
- APN contraseña: claro
- Nombre de dispositivo: ZTE

Los que se agregan en el siguiente comando, que inicializa el script UMTSKEEPER, los parámetros listados líneas arriba, se colocan en las zonas en negrita del comando.

```
- sudo ./umtskeeper --sakisoperators "USBINTERFACE='0'
OTHER='USBMODEM' USBMODEM='ID del modem' SIM_PIN='1234'
APN='CUSTOM_APN' CUSTOM_APN='APN de operadora'
APN_USER='APN usuario' APN_PASS='APN contraseña'" --
sakisswitches "--sudo --console" --devicename 'ZTE' --log --nat 'no'
```

Quedando de la siguiente manera:

```
- sudo ./umtskeeper --sakisoperators "USBINTERFACE='0'
OTHER='USBMODEM' USBMODEM='19d2:0031'
APN='CUSTOM_APN' CUSTOM_APN='claro.pe' APN_USER='claro'
APN_PASS='claro'" --sakisswitches "--sudo --console" --devicename 'ZTE'
--log --silent --monthstart 8 --nat 'no' --httpserv &>>
/home/pi/umtskeeper/error.log &
```

Ejecutamos y vemos que funciona correctamente.

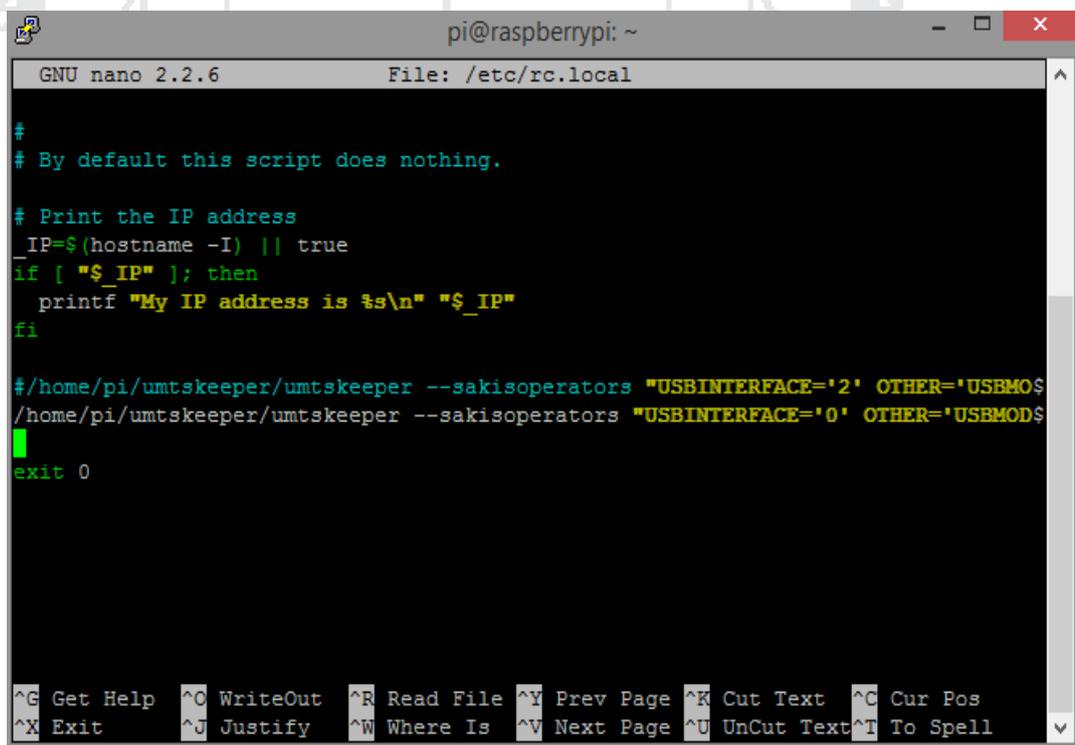
Finalmente, este script (umtskeeper) debe inicializar al arrancar el sistema operativo por lo que agregaremos el comando descrito anteriormente en el archivo rc.local, (todos los comandos agregados en este archivo se ejecutaran después de que el sistema ha iniciado.), al comando descrito se agregara la ubicación del script que se requiere inicializar, el comando quedaría así:

- ```
- /home/pi/umtskeeper/umtskeeper --sakisoperators "USBINTERFACE='0'
OTHER='USBMODEM' USBMODEM='19d2:0031'
APN='CUSTOM_APN' CUSTOM_APN='claro.pe' APN_USER='claro'
APN_PASS='claro'" --sakisswitches "--sudo --console" --devicename 'ZTE'
--log --silent --monthstart 8 --nat 'no' --httpsver &>>
/home/pi/umtskeeper/error.log &
```

Agregamos ese comando en el archivo rc.local como muestra la **Figura 55**.

Guardamos y reiniciamos, y debería conectarse automáticamente a internet.

- ```
- sudo nano /etc/rc.local
```



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/rc.local
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
# /home/pi/umtskeeper/umtskeeper --sakisoperators "USBINTERFACE='2' OTHER='USBMOD$
/home/pi/umtskeeper/umtskeeper --sakisoperators "USBINTERFACE='0' OTHER='USBMOD$
exit 0
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura 55: Archivo rc.local.

Elaboración: Propia.

3.10.6 Instalación y configuración de VPN

El VPN (virtual private network) o red privada virtual es una tecnología de red que permite la comunicación entre dos o más computadoras a través de internet simulando una red local, una de las aplicaciones más conocidas es Hamachi que permite

crear redes virtuales, aunque es de pago, permite la conexión de hasta 5 computadoras gratuitamente.

3.10.6.1 Instalación de hamachi.

Para la instalación de Hamachi crearemos una carpeta donde se guardaran los archivos correspondientes, previamente instalaremos los paquetes LSB y LSB-CORE (Linux Standard Base) que son un pre requisito para la instalación de Hamachi (Gomez, 2015) ejecutamos los siguientes comandos:

- sudo apt-get install --fix-missing lsb lsb-core
- mkdir hamachi
- cd hamachi
- sudo wget https://secure.logmein.com/labs/logmein-hamachi-2.1.0.139-armel.tgz

Este último comando descarga hamachi 2.1.0.139 para procesadores ARM

- sudo tar xvf logmein-hamachi-2.1.0.139-armel.tgz
- ls
- cd logmein-hamachi-2.1.0.165-armel
- sudo ./install.sh

```

pi@raspberrypi: ~/hamachi/logmein-hamachi-2.1.0.165-armel
pi@raspberrypi:~/hamachi $ cd logmein-hamachi-2.1.0.165-armel
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ ls
CHANGES  dnssup      hamachi-init  LICENSE  Release_notes.rtf
dnssdown  hamachid    install.sh    README   uninstall.sh
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ ^C
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ sudo ./install.sh
Copying files into /opt/logmein-hamachi ..
Creating LogMeIn Hamachi symlink ..
Creating TunTap ..
mknod: '/dev/net/tun': File exists
Installing LogMeIn Hamachi service ..
Traceback (most recent call last):
  File "/usr/lib/lsb/install_initd", line 46, in <module>
    headers = initdutils.scan_initfile(initfile)
  File "/usr/lib/lsb/initdutils.py", line 79, in scan_initfile
    inheaders = RFC822Parser(strob=headerlines)
  File "/usr/lib/lsb/initdutils.py", line 24, in __init__
    fileob = StringIO(strob)
TypeError: initial_value must be unicode or None, not str
Starting LogMeIn Hamachi service ..
Starting LogMeIn Hamachi VPN tunneling engine logmein-hamachi
starting - success
LogMeIn Hamachi is installed. See README for what to do next.
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $

```

Figura 56: Hamachi 2.1.0.139 correctamente instalado.

Elaboración: Propia.

Ya solo queda iniciar sesión y crear un nombre de usuario (Nick name) y conectarse a una red, por razones de seguridad solo nos conectaremos una red de prueba cuyos parámetros son:

- Id de red : raspiPC
- Contraseña: 12345678

Continuamos con los siguientes comandos:

- sudo hamachi login
- sudo hamachi set-nick **raspi Ubinas**
- sudo hamachi do-join **raspiPC** **12345678**

```

pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ sudo hamachi
version      : 2.1.0.165
pid          : 10030
status       : offline
client id    :
address      :
nickname     :
lmi account  :

pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ sudo hamachi login
Logging in ..... ok
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ sudo hamachi set-nick raspi_Ubinas
Setting nickname .. ok
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ hamachi do-join raspiPC 12345678
You do not have permission to control the hamachid daemon.
You can run 'hamachi' as root, or you can add your login name to the file
'/var/lib/logmein-hamachi/h2-engine-override.cfg'
and restart the daemon with
sudo /etc/init.d/logmein-hamachi restart
Example:
Ipc.User      <login name>
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $ sudo hamachi do-join raspiPC 123456
Joining raspiPC .. ok
pi@raspberrypi:~/hamachi/logmein-hamachi-2.1.0.165-armel $

```

Figura 57: Conexión de Raspberry PI a VPN exitosa.

Elaboración: Propia.

Finalmente es necesario que Hamachi se inicie automáticamente al encender la Raspberry pi con los siguientes comandos:

- sudo chmod 755 /etc/init.d/logmein-hamachi
- sudo /etc/init.d/logmein-hamachi start
- sudo update-rc.d logmein-hamachi defaults
- sudo reboot

3.10.7 Realizar script de envío de datos por VPN

El siguiente paso consiste en realizar un script que permita transferir la información generada por el prototipo hasta un servidor local, para ello es necesario contar con un cliente FTP en la Raspberry PI 3B+ que se comuniquen con un servidor FTP local, este prototipo nos permite usar dos medios de transferir información, una es usando una red ethernet cableada o inalámbrica sin internet, la que se hace usando su puerto ethernet, y la otra es usando un modem USB GSM/3G que se conecte a internet mediante una operadora de telefonía celular cualquiera, haciendo uso de una VPN, o red privada virtual. Las condiciones que debe cumplir esta transferencia son:

- La transferencia de información debe ser periódica y configurable.
- La información en el servidor local debe ser reflejo de la información almacenada en la memoria de la Raspberry PI 3 B+.
- Solo se debe transferir la información faltante en el servidor local, esto con el fin optimizar el ancho de banda que en ambas formas de transferencia es limitada y en caso se use modem USB se tiene un paquete de datos limitado por la operadora.

Para lograr este objetivo necesitamos instalar una pequeña aplicación llamada LFPT (cliente FTP), que es un programa para transferencia de archivos que soporta los protocolos ftp, http, sftp, fish, torrent, así como Bash, utiliza los mismos comandos propios de Linux, además cuenta con comandos para realizar sincronización de archivos y backup de archivos, y es capaz de enviar varios archivos en paralelo (Alvarez, 2016). Entonces instalamos la aplicación con el siguiente comando:

- `sudo apt-get install lftp`

Los comandos necesarios para hacer la transferencia de datos se harán en bash para ello crearemos un nuevo archivo con extensión “.sh” pero antes necesitaremos conocer la dirección IP del servidor local así como el nombre de usuario y contraseña, que nos brindara el administrador de la red del servidor donde queremos conectarnos.

- `cd scripts`
- `nano upload.sh`

Y escribimos los siguientes comandos

```
#!/bin/bash
```

```
# definición de parámetros para LFTP
```

```
HOST='25.109.102.144' #Escribimos la dirección IP del servidor local en la VPN.
```

USER='new_user' # Usuario que se asigna en el servidor local.

PASS='12345678' # Contraseña que se asigna en el servidor local.

TARGETFOLDER='/DATA' #Carpeta que se cuyo contenido se enviara al servidor local.

SOURCEFOLDER='/home/pi/DATAPGR' #Directorio de la carpeta a copiar.

lftp -f " # inicia las líneas de comando

open \$HOST # Abre la conexión con el servidor local.

user \$USER \$PASS # Introduce Usuario y contraseña.

lcd \$SOURCEFOLDER # introduce directorio.

mirror --reverse --delete --verbose \$SOURCEFOLDER \$TARGETFOLDER

#mirror es un comando que hace copias de respaldo entre dos estaciones remotas como si #de un espejo se tratara, y los parámetros

#reverse: indica que se quiere hacer una actualización de los archivos remotos.

delete: Indica que se eliminara los archivos del servidor remoto que no se encuentren en el #servidor local.

bye # Finaliza la transferencia

"

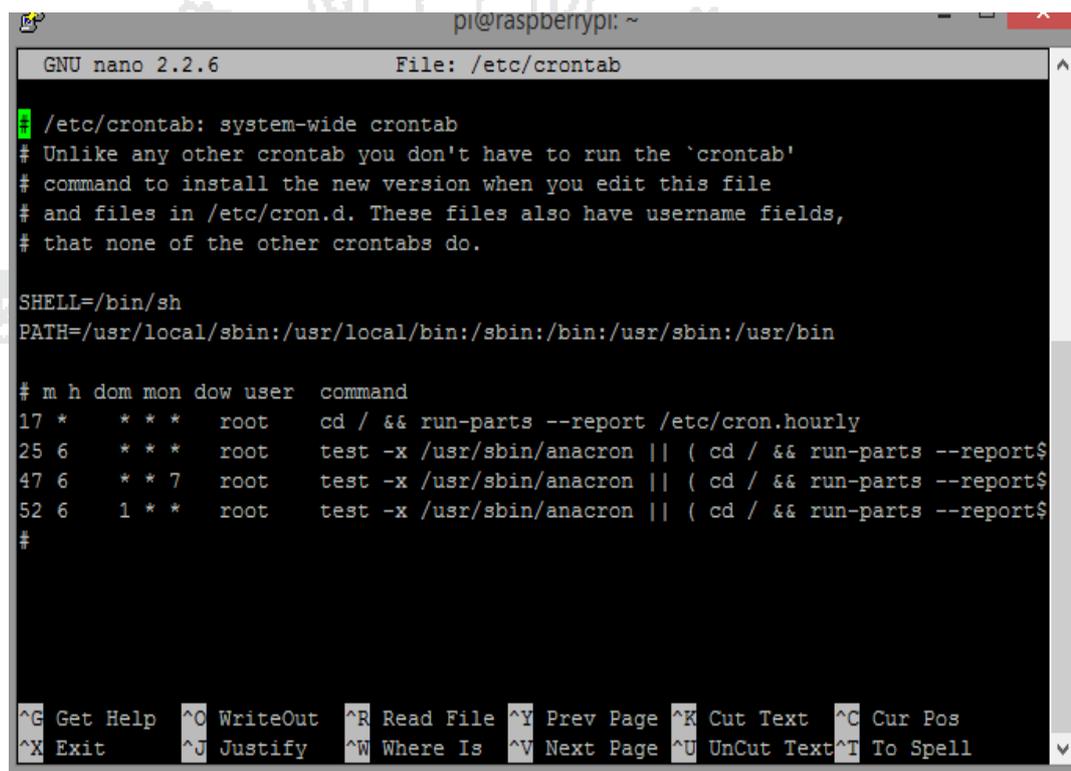
Más información sobre este cliente FTP se puede obtener en el sitio web del mismo (<http://lftp.tech/lftp-man.html>)

Este script transfiere los datos que se vayan generando en el prototipo y verifica que solo se envíen los datos faltantes en el servidor remoto, mas no hace esta tarea periódicamente, para esto recurriremos a una herramienta propia de Linux y también

presente en la Raspberry llamada CRON, que es un programa que se sirve para ejecutar comandos o scripts a una hora y fechas específicas (Gonzales, S.F).

Para manejar cron, modificaremos directamente el archivo `/etc/crontab`, el formato que usa cron para programar la ejecución de comandos o scripts se muestra en la **Figura 58**

- `sudo nano /etc/crontab`



```

GNU nano 2.2.6      File: /etc/crontab

/etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report$
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report$
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report$
#
^G Get Help  ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit     ^J Justify  ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Figura 58: Formato para programar tareas en Cron.

Elaboración: Propia.

Como vemos cada línea consta de 7 partes que de izquierda a derecha son: minutos, hora, día de mes, mes, día de la semana, usuario y comando o script, el símbolo “*” significa en cada campo significa que la acción se realizara en todo momento, es decir si estuviera en el campo de minuto significa que la tarea se realizara cada minuto, si estuviera en horas la tarea se realizaría cada hora. Mostraremos dos ejemplos, uno que hará que el script se ejecute cada hora y otro donde el script se ejecute cada día.

Para que se ejecute cada hora, digamos en el minuto 5 de cada hora, el comando seria el siguiente:

- 05 * * * * bash /home/pi/scripts/upload.sh

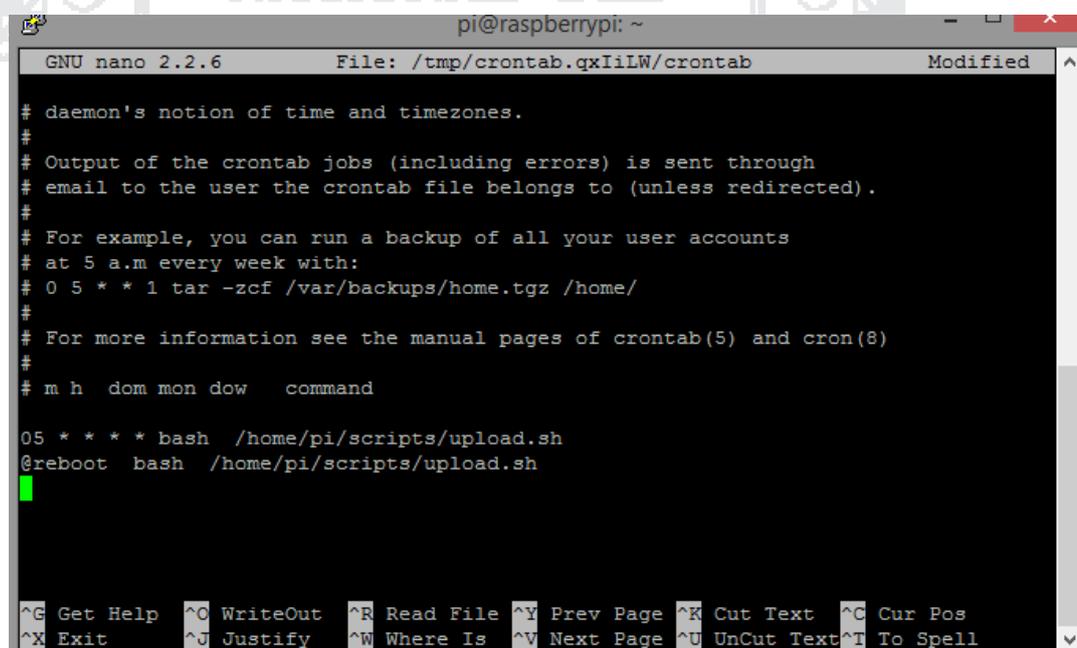
Y para que se ejecute una vez al día, digamos a las 10:00 PM de cada día, el comando sería el siguiente:

- 00 22 * * * bash /home/pi/scripts/upload.sh

Por tanto, procedemos editando la lista de tareas de cron con el comando

- Sudo crontab -e

Y una vez dentro del archivo copiamos el comando que hará que se ejecute el script cada hora, la **Figura 59** muestra cómo quedaría.



```

pi@raspberrypi: ~
GNU nano 2.2.6 File: /tmp/crontab.qxIiLW/crontab Modified
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
05 * * * * bash /home/pi/scripts/upload.sh
@reboot bash /home/pi/scripts/upload.sh
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Figura 59: Crontab programado para ejecutar upload.sh cada hora.

Elaboración: Propia.

3.11 CALIBRACION E INSTALACIÓN DEL PROTOTIPO

3.11.1 Calibración.

El instrumental utilizado fue el siguiente

- Multímetro: Fluke 179

- Laptop
- Batería de iones de litio de 3.7 v
- Potenciómetro de 10k y 100K

Como bien sabemos el resultado que entrega un ADC es un número que en nuestro caso está comprendido entre 32767 y -32766 , el cual está relacionado con el valor medido de manera lineal, para que nos muestre el resultado que se busca, se debe establecer una ecuación lineal que relacione el valor entregado por el ADC y el valor que realmente nos interesa por lo tanto el procedimiento es el siguiente:

- Preparar en una mesa el multímetro, el prototipo y la batería con los potenciómetros conectados en serie como muestra la **Figura 60** y en otra mesa la computadora, conectarse a la Raspberry PI 3 b+ desde la laptop, detener el script que toma las muestras, para iniciarla manualmente un script que viene en los ejemplos de la librería del AD1115 y modificamos los parámetros de ganancia=1 y canal=0 que son los que usara normalmente nuestro script:
 - sudo /etc/init.d/autoinicio_ADC stop
 - cd Adafruit_Python_ADS1x15
 - cd examples
 - sudo python differential.py

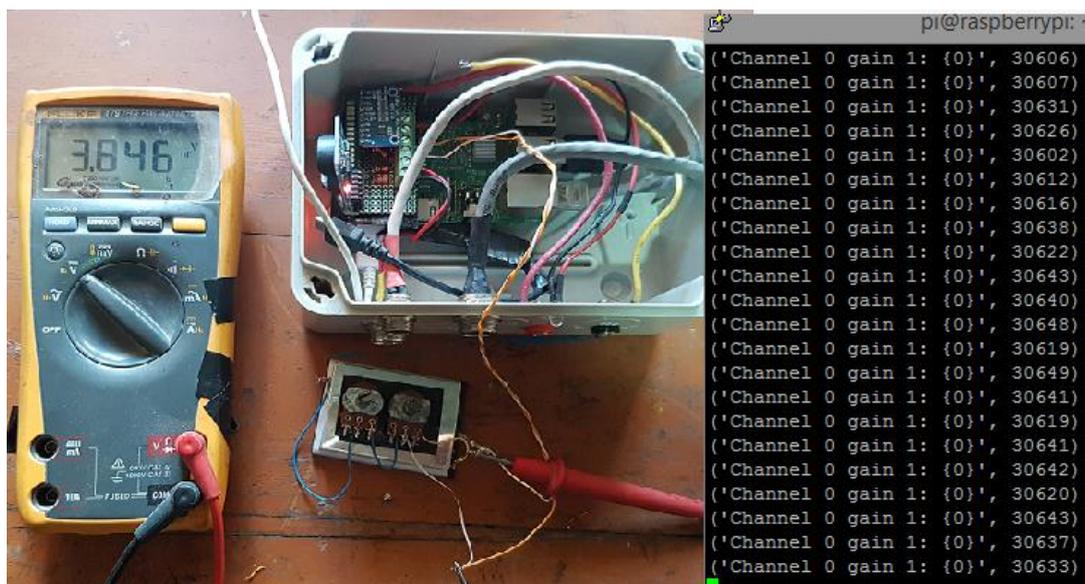


Figura 60: Calibrando el prototipo.

Elaboración: Propia.

- con el procedimiento anterior obtenemos la **Tabla 9** haciendo uso de Excel obtenemos las gráficas **Figura 61** que muestra que el resultado del ADC es lineal,

Tabla 9: Relación entre voltaje y valores de ADC.

Voltaje V	Valor ADC
3.844	30570
0.044	24342
2.32	18554
1.522	12196
0.771	6220
0.327	2656
0.278	2237
0.184	1472
0.144	1164
0.092	721
0.057	441
0.024	191
0.01	76
0.005	41
0.001	7

Elaboración: Propia.

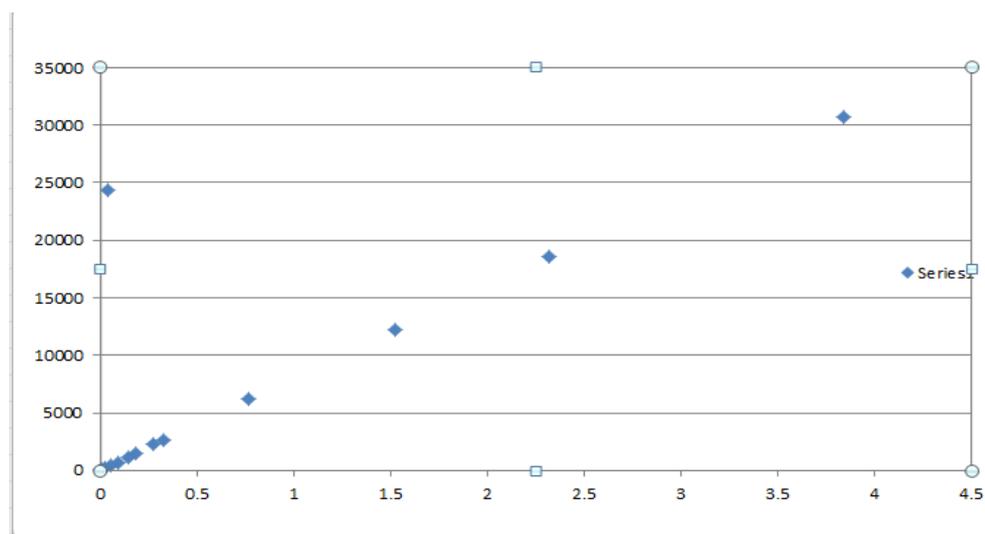


Figura 61: Grafica a rango completo.

Elaboración: Propia.

- luego obtenemos la ecuación lineal que relación ambos valores del obtendremos una constante K que se usara en el script escrito en Python que nos dará finalmente el valor deseado.

$$V = \frac{3.844}{30570} ADC \tag{5}$$

- Finalmente comprobamos que la calibración está hecha moviendo el valor del potenciómetro a un valor aleatorio y comparamos los resultados, estos se muestran en la **Figura 62**. Este procedimiento también aplica para voltajes negativos, pues recordemos que el ADC tiene entrada diferencial.

$$E_r = 1.018 \%$$

3.11.1.2 Precisión del prototipo.

La precisión es la proximidad entre las indicaciones o los valores medidos obtenidos en mediciones repetidas de un mismo objeto, bajo condiciones especificadas. Por tanto calcularemos la desviación estándar con los valores de la **Tabla 10** y obtenemos:

$$D=0.00119634$$

Lo que significa que la variación es muy pequeña entre medida y medida por tanto tiene buena precisión.

3.11.2 Instalación del prototipo

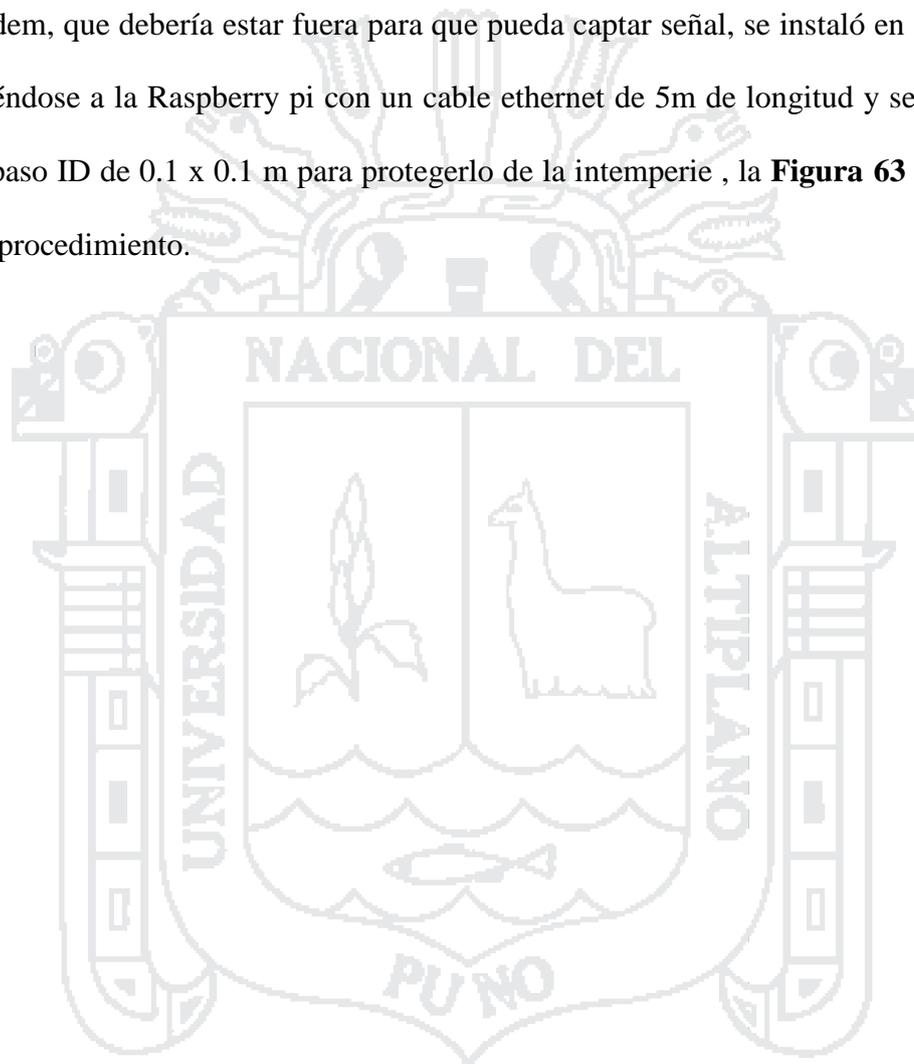
La instalación se realizó el 20 del mes agosto del año 2015 y estaba sujeta a ciertas condiciones

- Debería ser de bajo coste
- Debería tomar máximo un día.
- No debería requerir mucho personal.
- Los más seguro posible

Todo esto debido al lugar de instalación pues desde la localidad de Ubinas toma 1 hora en camioneta y de ahí se debe caminar cuesta arriba por 3 horas, por tanto la logística no debería grande, el prototipo en campo cuenta, como fuente de alimentación, con un sistema fotovoltaico que consta de un panel de 85Watt. Un regulador Sunsaver de 10 y batería de 35 AH, no se detallara masen este debido a que no es parte de la investigación y se utilizó lo que se tenía a mano.

Como contenedor se usó un cubo de plástico de los que se usan para transporte de aceite, el personal requerido fue de dos ayudantes de la zona, cuyo trabajo fue cavar los

agujeros donde estaría la cubeta y los electrodos, 2 agujeros de 1m x 0.3m de radio para los electrodos y un agujero de 1 x 0.8 x 0.5 m para la cubeta, y una zanja de 100 metros de largo por 0.3 m de profundidad con orientación radial al cráter del volcán que unía los agujeros donde se encontraban los electrodos, el panel solar se anclo en una roca y el modem, que debería estar fuera para que pueda captar señal, se instaló en una parte alta uniéndose a la Raspberry pi con un cable ethernet de 5m de longitud y se usó una caja de paso ID de 0.1 x 0.1 m para protegerlo de la intemperie , la **Figura 63** muestra parte del procedimiento.



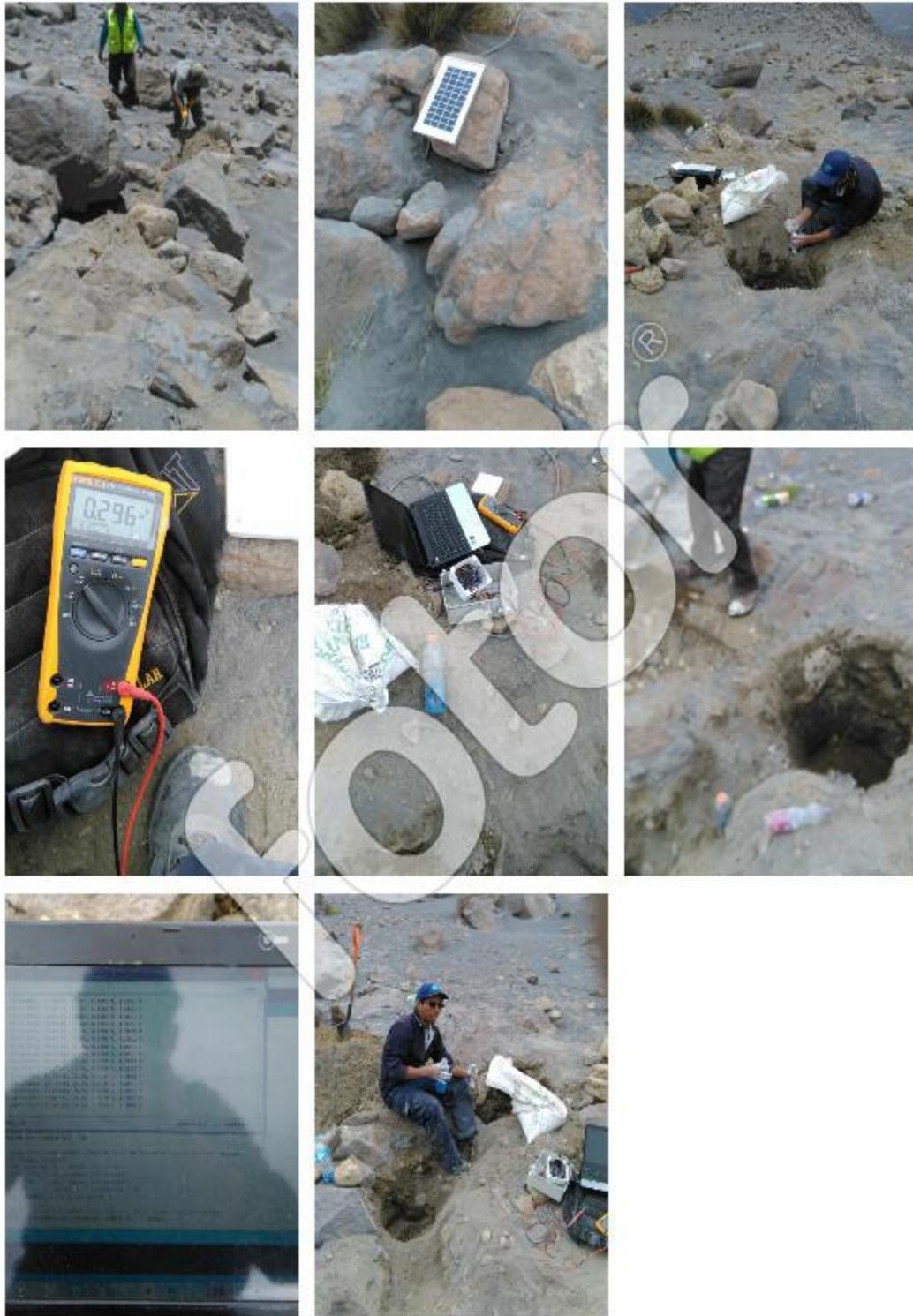


Figura 63: Instalación del prototipo en el volcán Ubinas.

Elaboración: Propia.



4.1 ANALISIS E INTERPRETACION DE RESULTADOS

En este capítulo se mostrara los resultados de la investigación, iniciando con los resultados de la aplicación, que nos indicara la utilidad del prototipo en el monitoreo del volcán Ubinas, haciendo una comprobación de anomalías registradas por los métodos de monitoreo visual y sísmico en un mismo periodo de tiempo, seguidamente mostraremos los resultados del diseño del prototipo y sus características técnicas y una comparación entre los costos de implementación e instalación del prototipo y los costos si se hubiera usado un equipo profesional.

4.2 RESULTADOS DE LA APLICACIÓN

El prototipo estuvo instalado durante todo un mes de prueba en el mes de agosto del 2015, las muestras se tomaron cada 30 segundos. Los resultados obtenidos se muestran en la **Figura 64** muestra las lecturas de los días comprendidos entre 20 de agosto y el 21 de setiembre.



Figura 64: Lecturas correspondientes a los días comprendidos entre el 20/08/2015 y el 21/09/2015.

Elaboración: Propia.

Donde se puede observar un constante ascenso de voltaje que podría responder a un aumento de temperatura en el edificio volcánico, esto lo podemos corroborar notando que los picos más altos de los rizados se corresponden a las horas donde la incidencia del sol es mayor, y repentinamente una notoria anomalía entre los días 29 de agosto y 31 de agosto, iniciándose el día **29 de agosto a las 13:09:00** con un voltaje de **0.29 V**, a partir de donde el voltaje crece repentinamente alcanzando un máximo de **0.405 V a las 16:55:00** del mismo día, permaneciendo en ese valor por unos minutos, y empieza a disminuir aunque con una pendiente menos pronunciada, esto se debería al descenso de la temperatura del terreno debido a que ya se va haciendo noche y respondería a las variaciones de temperatura debido a la transición de día a noche. Lo que nos sugiere que si no existieran los rizados debido a la transición de día y noche, el descenso del voltaje debería ser continuo hasta el punto donde se vuelve hacer estable nuevamente.

Para comprobar la utilidad del prototipo es necesario revisar los datos generados por otros métodos de monitoreo volcánico en busca de anomalías que hayan sido detectadas durante el periodo de tiempo donde se dieron los cambios más relevantes de las medidas de potencial espontáneo, para este propósito utilizaremos los datos del monitoreo visual y el monitoreo sísmico.

4.2.1 Correlación con los datos obtenidos por el monitoreo visual.

4.2.1.1 Observación del día 29/08/2015

El monitoreo visual trata de la toma de imágenes del volcán cada 30 segundos, la **Figura 65** muestra una serie de imágenes correspondientes al monitoreo visual del volcán Ubinas del día 29/08/2015, si nos centramos en las imágenes previas a la hora donde la anomalía de potencial se inicia, es decir a las 13:09:00 (ver **Figura 64**), y

vemos la **Figura 65e**, no existen cambios aparentes de hecho es el comportamiento de días anteriores y solo se observa una pequeña humareda a las 8:46:00, **Figura 65b**, lo que no resulta importante.

Si ahora observamos la imagen que corresponde con la hora donde la anomalía de potencial alcanza su pico máximo es decir las 16:55:00 (ver **Figura 64**), y vemos la **Figura 65h**, tampoco observamos nada relevante, de hecho el volcán se muestra más calmado que en horas de la mañana.

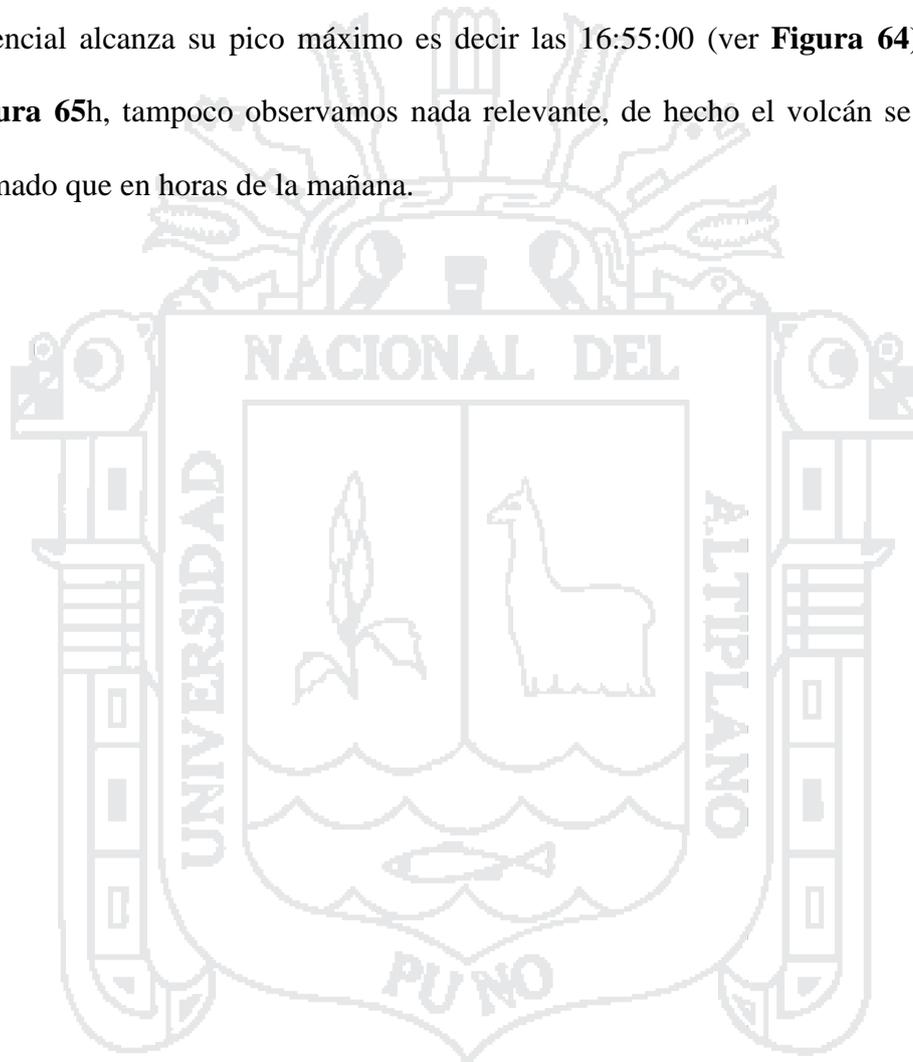




Figura 65: Monitoreo visual 29/08/2015.

Fuente: Recuperado de la Base de datos monitoreo visual OVI.

4.2.1.2 Observación del día 30/08/2015

Como vimos las imágenes del día **29/08/2015** no nos ofrece nada relevante, sin embargo en el amanecer del día 30, **Figura 66a**, el volcán muestra una emisión de ceniza relativamente grande pero no podemos precisar la hora en que se inició la emisión debido a la falta de luz. La emisión de ceniza es constante hasta a aproximadamente las **9:27:00**, **Figura 66d**, donde ya no hay emisión de ceniza salvo algunas emisiones leves de gases las siguientes horas, pero la emisión de ceniza vuelve a darse a par de las **11:47:30**, **Figura 66f**, manteniéndose hasta finalizar el día.





Figura 66: Monitoreo visual 30/08/2015.

Fuente: Recuperado de la base de datos monitoreo visual OVI.

Finalmente de la correlacion entre las lecturas del prototipo y las imágenes del monitoreo visual sugiere que pueden haber cambios en el campo eléctrico de la superficie del edificio volcánico horas antes de que estos se hagan visibles en forma de una emisión o incluso erupción del volcán Ubinas, sin embargo no nos permite dar con precisión el lapso de tiempo en horas, desde que se ven anomalías en el potencial espontaneo y el momento en que las emisiones de ceniza inician, esto debido a que la falta de luz nos lo impide.

4.2.2 Correlacion con los datos obtenidos con el monitoreo sísmico

Para realizar esta correlacion recurrimos a la base de datos de monitoreo sísmico del OVI, en forma de helicorder que muestra el comportamiento de la señal sísmica en su eje Z,

4.2.2.1 Observación del helicorder del día 29/08/2015

Debemos tomar en cuenta que en el helicorder las lecturas están hechas en tiempo coordinado universal (UTC), para poder hacer la comparaciones en el helicorder deberemos sumar 5 horas a la hora local es decir a la hora de las anomalías de las lecturas de potencial le sumaremos 5 horas, así pues tenemos:

- Hora del punto más bajo de la anomalía de potencial, ver **Figura 64**.
13:09:00 LOCAL= 18:09:00 UTC
- Hora del punto más alto de la anomalía de potencial, ver **Figura 64**.
16:55:00 LOCAL = 21:55:00 UTC

Con esto vemos la **Figura 67** los puntos marcados en círculos muestran las horas de los putos bajo y alto respectivamente, de la anomalía de potencial, donde no se ve nada anormal de hecho el helicorder no muestra ninguna anomalía durante todo el día 29/08/2015.

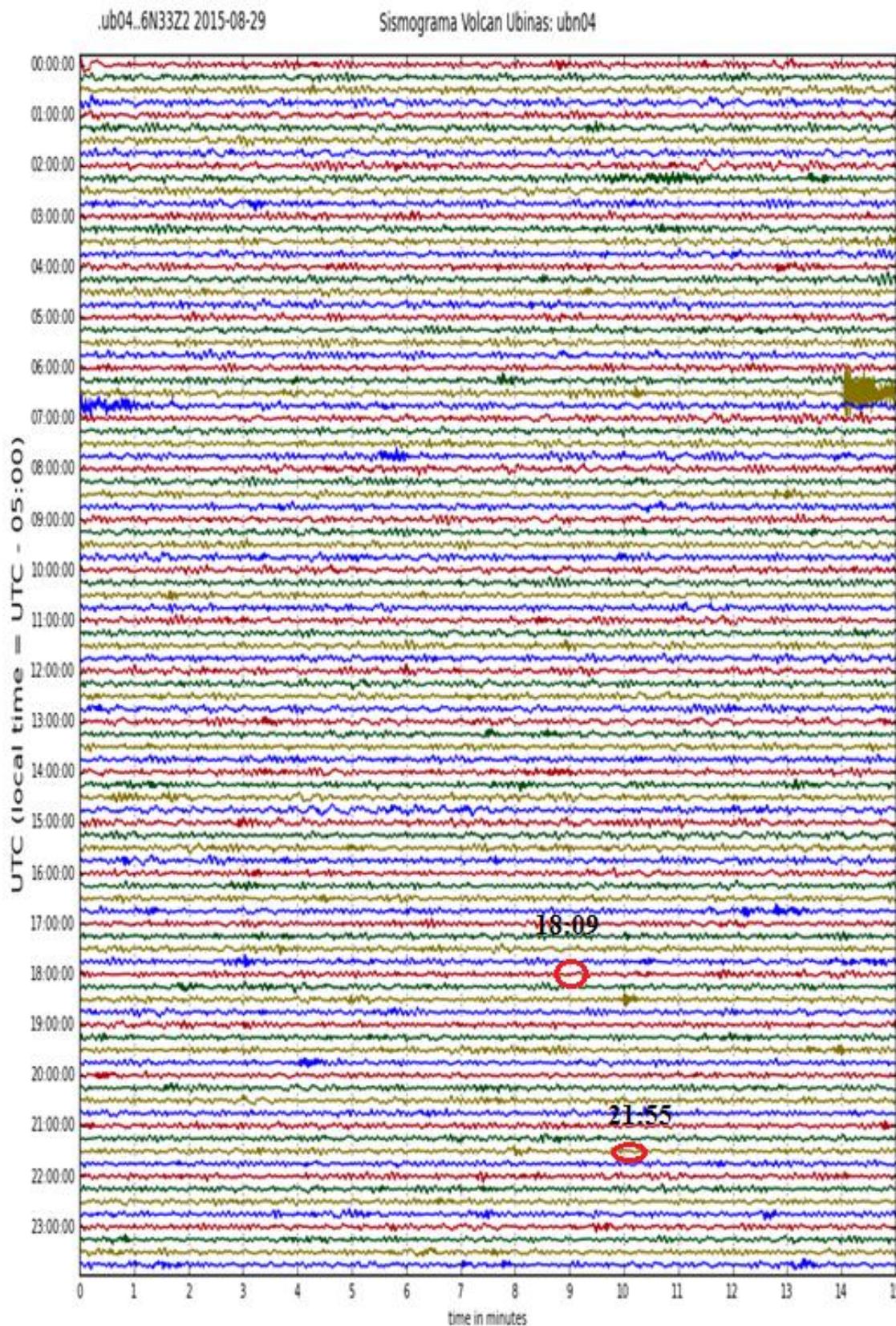


Figura 67: Helicorder del 29/08/2015.

Fuente: Recuperado de base de datos monitoreo sísmico OVI.

4.2.2.2 Observación del helicorder del día 29/08/2015

Si observamos la **Figura 68** podemos notar que existe una anomalía que inicia a las **9:22 UTC (4:22 hora local)** y la anomalía se hace más grande y continua a las **9:34 UTC (4:34 hora local)**, lo cual es interesante pues nos podría indicar el momento en que inicio las emisiones de ceniza que se pueden ver en la **Figura 66**, otro aspecto interesante que debemos notar es la diferencia de horas desde el inicio de la anomalía de potencial y la hora en que iniciaron las anomalías en el helicorder y por ende en las imágenes del monitoreo visual.

Para realizar la operación tomaremos en cuenta la hora del inicio de la anomalía en el potencial y la hora en que inicia la anomalía en el helicorder, tenemos que hay una diferencia de 15 horas y 13 minutos, y si tomamos en cuenta el punto más alto de la anomalía de potencial espontaneo es decir las 16:55:00 del día 29/08/2015 (ver **Figura 64**), tenemos una diferencia de 11 horas con 27 minutos, para efectos prácticos tomaremos este último valor como la diferencia en tiempo de la ocurrencia de las anomalías.

Esta diferencia se puede explicar del siguiente modo, inicialmente el conducto volcánico se encuentra obstruido y al ascender el magma esta hace que en partes internas del volcán la temperatura aumente, con este aumento de temperatura el agua del sistema hidrotermal del volcán empieza a evaporar y aumenta la presión dentro del volcán, este vapor de agua empieza a salir por filtraciones de arena que hacen que la temperatura del subsuelo del volcán aumente causando las anomalías de voltaje, este proceso de avance y aumento de presión y temperatura hace que en un momento dado el conducto se habrá y empiece una emisión de ceniza.

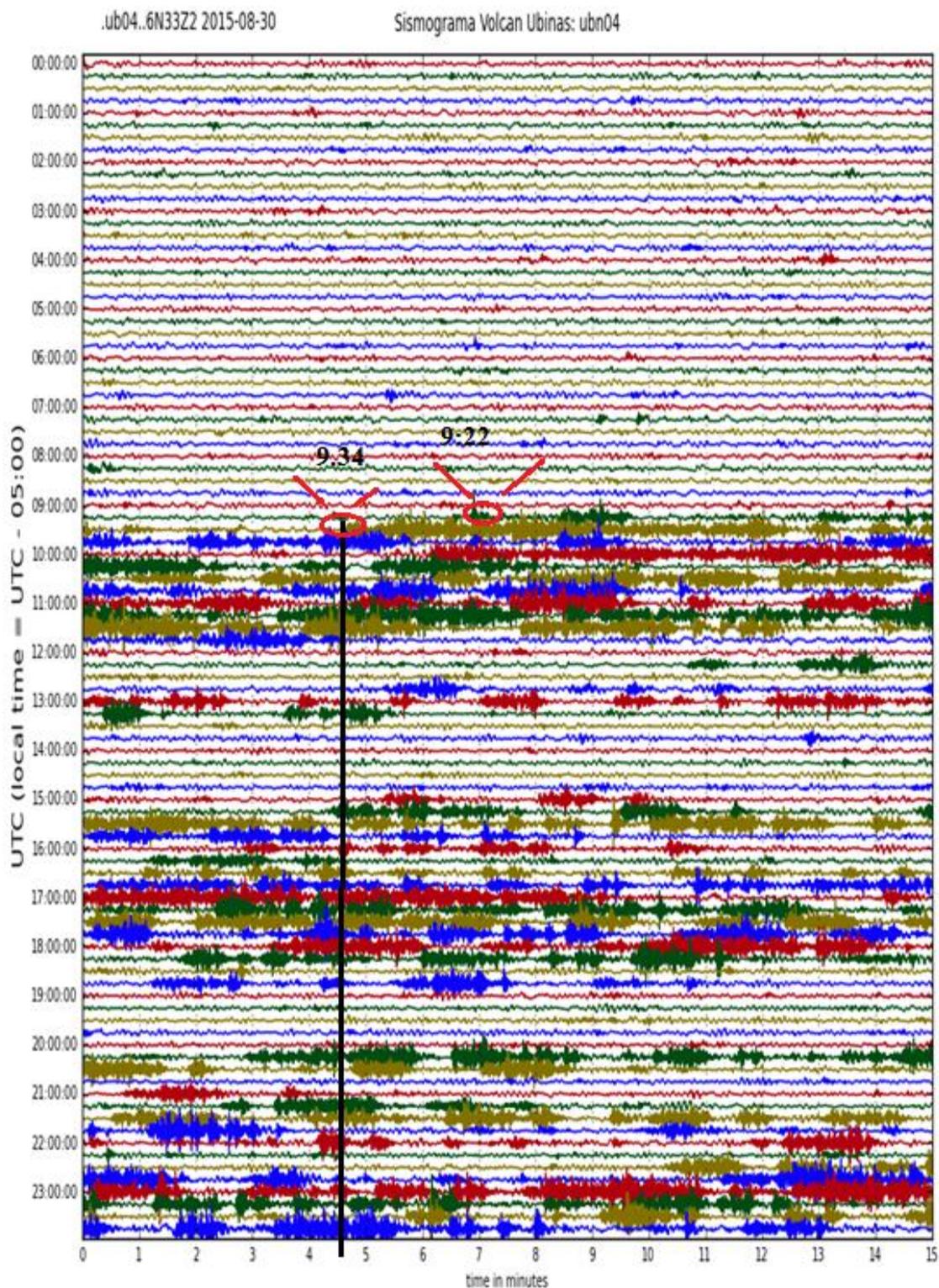


Figura 68: Helicorder del 30/08/2015.

Fuente: Recuperado de la base de datos monitoreo sísmico OVI.

Con todo esto podemos concluir que el prototipo funciona, y las lecturas hechas forman un indicio de que en realidad existen cambios a nivel de potencial eléctrico del suelo en el edificio volcánico previos a una erupción incluso con hasta 11:27 horas de anticipación, si ese fuera el caso estaríamos ante un método que aportaría mucho al monitoreo del volcán Ubinas ya que nos acercaría un poco más al tan anhelado objetivo de predecir la erupción de un volcán, lo que permitirá dar alertas tempranas favoreciendo mucho a la población que vive alrededor, y las actividades económicas que podrían ser afectadas por una erupción volcánica.

El panorama que nos ofrece estos resultados con alentadores sin embargo no olvidemos que es solo un indicio, y hacen falta muchos más indicios para que puedan conformar una prueba.

Finalmente la comparación hecha entre los datos obtenidos por el prototipo y los datos obtenidos por el monitoreo visual y sísmico, nos muestra que la información obtenida con el prototipo es de utilidad en el monitoreo del volcán Ubinas. Y nos muestra que con este método podemos esperar un evento volcánico con hasta 11 horas y media de anticipación.

4.3 RESULTADOS DEL DISEÑO E IMPLEMENTACION DEL PROTOTIPO.

La investigación inicio con busca de información sobre plataformas y tarjetas de desarrollo que cumplieran con los requerimientos que precisan este tipo de mediciones en campo, se exploró en el campo de las tarjetas de desarrollo, ADC's, RTC, reguladores de voltaje. Para dar con el más apropiado hardware, y se exploró también las opciones existentes para transmisión de información desde lugares totalmente inhóspitos hasta las instalaciones del observatorio del ingemmet para que los datos

fueran estudiados rápidamente y así la investigación que se viene haciendo en torno a este fenómeno vaya más aprisa.

Por tanto el resultado de todo ese estudio y exploración se plasma en un prototipo de medición de potencial espontaneo con las siguientes características:

- Sensibilidad : $< 1\text{mV}$
- Exactitud: con un error relativo aprox. de un 1%
- Precisión: con una desviación de 0.00011v
- Impedancia de entrada: $> 10\text{Mohm}$
- Resolución: 16 bits
- Rango de entrada programable.
- Entrada diferencial.
- Almacenamiento de datos: $>25\text{GB}$
- Con capacidad de conexión a internet vía red GPRS/3G
- Con capacidad de conexión a una red LAN.
- Periodo programable desde 1 segundo

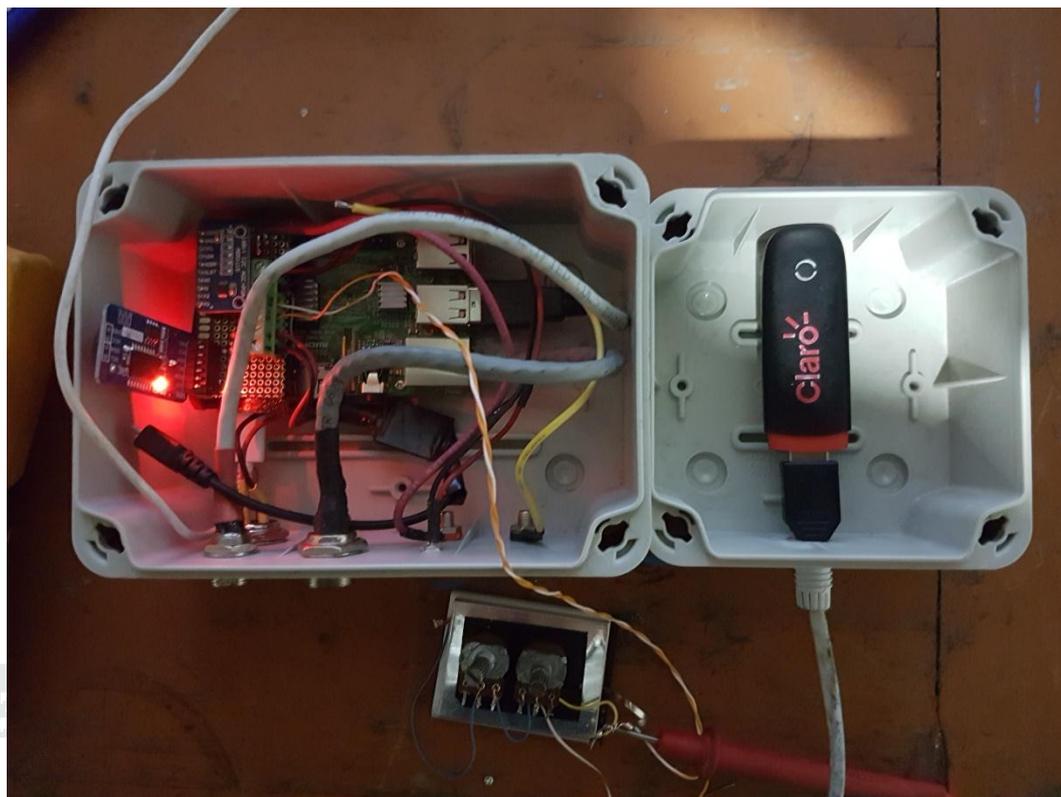


Figura 69: Prototipo de medición de potencial espontáneo.

Elaboración: Propia.

Sin embargo para comprobar nuestra hipótesis compararemos el costo de implementación e instalación si utilizáramos un equipo profesional, para este caso utilizaremos de ejemplo el modelo de **Datalogger CR6** de la marca CAMPBELL, sugerido por especialistas franceses colaboradores de la institución, la **Tabla 11** muestra el costo individual del equipo, la institución tiene por política asegurar sus equipos caros pero estos seguros exigen ciertas condiciones de instalación para que se puedan hacer responsables de los costos si estos fueran robados o dañados de por rayos, del mismo modo sucede con la garantía del equipo, esto hace que los costos de instalación del equipo se incrementen mucho, pues se debe considerar toda una estructura metálica que provea seguridad ante robos cuyo costo ronda los S/.15000 y adicionalmente un sistema de puesta a tierra para protegerlo contra rayos cuyo costo es aproximadamente S/. 10000 esos costos solo son materiales, ahora se debe tomar en cuenta también la logística, personal y días de trabajo lo que implicaría una comisión exclusivamente para

esta tarea, solo en logística el costo ascendería en S/. 20000, con 5 días de trabajo, 7 ayudantes en campo, comisión de servicio y combustible. Los costos acá mencionados son aproximados pues dependiendo de la zona donde se pretenda instalar el equipo el costo puede aumentar mucho más. Además el Datalogger CR6 solo cuenta con un puerto ethernet lo que implicaría que este debería conectarse a la red LAN propia de la institución lo que significaría el uso de 2 tranceiver de los que tiene la institución (freewave FGR2 PE de 900MHz), lo que incrementaría aún más el costo y limitaría mucho las zonas de instalación.



Figura 70: Datalogger profesional CR6 de CAMPBELL.

Fuente: Recuperado de manual del CR6 de CAMPBELL.

Por otro lado el costo de instalación del prototipo es tan bajo porque para su instalación no requiere de una comisión exclusiva, y se puede realizar dentro de una comisión cuyo objetivo corresponda a otro método de monitoreo, y debido al poco personal necesario (dos ayudantes) y materiales necesarios la logística se cuenta desde el punto hasta donde llega la camioneta y de ahí la movilidad puede dedicarse a la realización de otras áreas con los otros profesionales encargados de su área.

La siguiente lista es de los materiales utilizados en la instalación

- Días de instalación: 1 día
- Personal: 3, dos ayudantes y mi persona.
- Herramientas requeridas: pico, pala, laptop, taladro
- Insumos: cubeta, cable 14awg, bentonita sódica, pernos de expansión
- Sistema fotovoltaico: 1 panel de 85Watt, regulador de 10, batería.
- Costo \approx S/ 1500

Finalmente la **Tabla 11** valida nuestra primera hipótesis específica pues nos muestra un resumen de los costos de equipamiento e instalación donde vemos que nuestro prototipo de es muy bajo coste.

Y la **Tabla 12** resume los costos de la tecnología de transmisión de las que pueden disponer cada equipo, tomado como costo inicial pues el servicio de telefonía móvil ira incrementándose mensualmente a razón de 30 soles mensual, y aun así es mucho más barato que la utilización de tranciver. Finalmente con esto validamos nuestra segunda hipótesis específica.

Tabla 11: Comparación de costo de equipo e instalación.

Item	Conectividad	Costo del equipo S/.	Costo de instalación S/.	Costo de software S/.	Costo total S/.
Prototipo con Rpi 3	LAN, GSM/GPRS/3G	320	1500	Libre	1820
Datalogguer CR6	LAN	12875*	45000*	3889*	61764*

* Los precios fueron cedidos por personal del OVI.

Elaboración: Propia.

Tabla 12: Comparación de costo de tecnología de transmisión.

Item	Equipo de transmisión	Costo del equipo S/.	Costo de servicio S/. mensual	Costo total S/.
Prototipo con Rpi 3	modem GSM/GPRS/3G	70	30	100
Datalogger CR6	Tranceiver freewave GPR2-PE 900 MHz x 2	15000*	0	15000*

* Los precios fueron cedidos por personal del OVI.

Elaboración: Propia.



CONCLUSIONES

PRIMERO: Como conclusión podemos afirmar que el prototipo funciona correctamente y gracias a la correlación con los datos de los métodos de monitoreo sísmico y visual **podemos decir este método nos puede ayudar a predecir algún evento volcánico con horas de anticipación, para este caso particular de hasta 11 horas y 27 minutos.**

SEGUNDO: El hardware utilizado para la realización del prototipo es mínimo, pues no consta de muchos componentes, y eficiente porque logra el objetivo a bajo coste, **la implementación significo un ahorro S/.12555 nuevos soles en equipo de medición y S/.43500 nuevos soles en costos de instalación. Haciendo un total de S/.56055 aproximadamente.**

TERCERO: La tecnología de telecomunicaciones usada es de muy bajo costo pues nos valemos de una red privada ya instalada por la que solo pagamos por el servicio. **Lo cual significo un ahorro de S/.14900 soles en costo de transmisión de los datos.**

RECOMENDACIONES

PRIMERO: Si se desea agrandar las capacidades del prototipo como entradas analógicas, se pueden agregar módulos por bus SPI, pues solo se pueden agregar hasta tres módulos más al puerto I2C, como el AD7755, que tiene características similares al usado en este prototipo.

SEGUNDO: Si se provee agregar más sensores, módulos, etc. Se debe contar con alimentación externa de 3.3v o una de 5 v pero se debe contar con conversores de estado para conectarlo a la Raspberry PI 3B+, pues su GPIO no serían capaces de abastecer esa carga y podría dañarse.

TERCERO: La mejor opción para la transmisión de datos, es siempre, la red dedicada si cuenta con una, pues las operadoras como claro o movistar siempre caen en zonas donde la cobertura es pobre.

CUARTO: Cuando esté trabajando con la Raspberry PI 3 B+, siempre mantenga orden en su área de trabajo, ya que la posibilidad de la ocurrencia de accidentes siempre está presente y podría dañar la Raspberry u otro componente

QUINTO: Se recomienda desarrollar a una aplicación web que te permita configurar los canales, rangos, IP, etc. Para que sea más fácil de usar por gente no entendida en el tema de Linux.

BIBLIOGRAFIA

- Ada, L. (13 de Setiembre de 2016). *Adafruit DS3231 Precision RTC Breakout*. Recuperado el 15 de Noviembre de 2016, de learn.adafruit: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ds3231-precision-rtc-breakout.pdf>
- Alvarez, M. (11 de enero de 2016). *FTP por línea de comandos avanzado LFTP*. Obtenido de DesarrolloWeb: <http://www.desarrolloweb.com/articulos/2319.php>
- Baptista Lucio , P., Hernadez Sampieri, R., & Fernandez Collado, c. (2006). *Metodologia de la investigacion*. Iztapalapa, Mexico: Infagon Web S.A.
- computing at school and BCS. (2012). *The raspberry pi, education manual*. Inglaterra.
- Doutel, F. (04 de abril de 2014). *Grabar la SD para tu Raspberry Pi no puede ser más sencillo con estas aplicaciones*. Recuperado el 25 de noviembre de 2016, de xatakahome: <https://www.xatakahome.com/trucos-y-bricolaje-smart/grabar-la-sd-para-tu-raspberry-pi-no-puede-ser-mas-sencillo-con-estas-aplicaciones>
- Earl, B. (04 de Octubre de 2016). *Adafruit 4-Channel ADC Breakouts*. Recuperado el 10 de Noviembre de 2016, de learn.adafruit: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-4-channel-adc-breakouts.pdf>
- Elias. (04 de Octubre de 2015). *UMTSkeeper: keep your UMTS/GPRS/GSM connection alive automatically*. Recuperado el 19 de noviembre de 2016, de mintakaconciencia: <http://mintakaconciencia.net/squares/umtskeeper/>
- Figuerola, J. (1974). *Tratado de geofisica aplicada*. Madrid: Sección de Publicaciones del Instituto Geológico de Minas.
- Floyd, T. L. (2006). *Fundamentos de sistemas digitales* . Madrid: PEARSON EDUCACIÓN S.A.

Gay, W. (2014). *Mastering the raspberry PI*. Toronto.

Gex, P. (1977). *la methode de polarisation spontanee*. lausanne.

Gomez, J. (14 de Noviembre de 2015). *Install Hamachi on a Raspberry Pi (Raspbian)*.

Obtenido de Joe Gomez: <https://www.jgomez.net/install-hamachi-on-a-raspberry-pi-raspbian/>

Gonzales Zuñiga, K. (2000). Metodo del potencial espontaneo aplicado al volcan Ubinas y metodos geoquimicos aplicados a la vigilancia volcanica y sismica. *Revista de trabajos de investigacion CNDG*, 47-58.

Gonzales, S. (S.F). *manual basico de CRON*. Recuperado el 24 de noviembre de 2016, de linux total: http://www.linuxtotal.com.mx/?cont=info_admon_006

GoogleEarth. (12 de Noviembre de 2016). *GoogleEarth*. Obtenido de GoogleEarth: <https://www.google.es/intl/es/earth/index.html>

googlemaps. (15 de Abril de 2016). *googlemaps*. Recuperado el 15 de abril de 2016, de googlemaps:

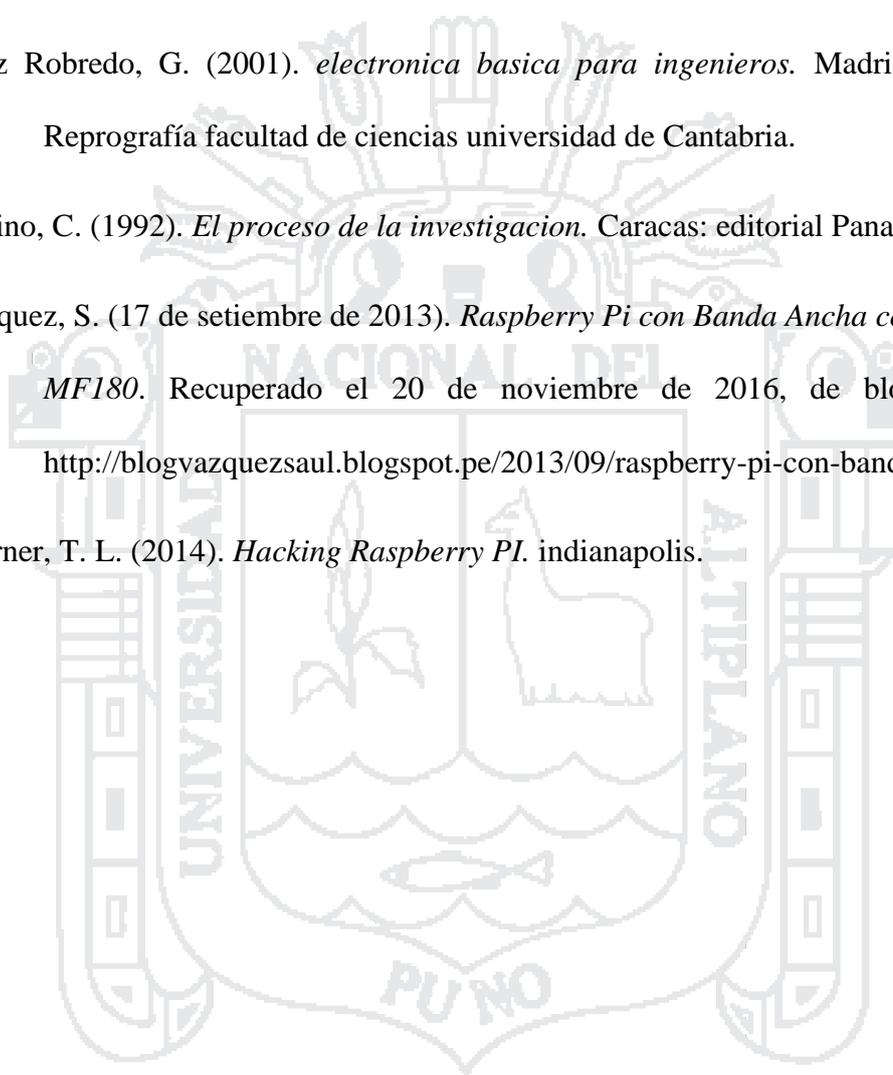
<https://www.google.com.pe/maps/place/Observatorio+Vulcanol%C3%B3gico+d+el+INGEMMET+-+OVI/@-16.3969108,-71.5490611,17z/data=!3m1!4m5!3m4!1s0x91424a5da0da88b3:0xfd9321404e45e0ac!8m2!3d-16.396916!4d-71.546867>

Gorkau. (20 de Diciembre de 2013). *Cómo ejecutar un programa automáticamente al arrancar la Raspberry Pi*. Recuperado el 2 de diciembre de 2016, de Ni idea de redes: <https://nideaderedes.urlansoft.com/2013/12/20/como-ejecutar-un-programa-automaticamente-al-arrancar-la-raspberry-pi/>

Huidobro, J. M. (2012). *comunicaciones moviles, sistemas GSM, UMTS, LTE*. RA-MA.

- Industries, A. (04 de Octubre de 2016). *adafruit 4 channel adc breakouts.pdf*.
Recuperado el 2 de Noviembre de 2016, de learn.adafruit.com: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-4-channel-adc-breakouts.pdf>
- Jan. (25 de marzo de 2016). *Adding a DS3231 Real Time Clock to the Raspberry Pi 3*.
Obtenido de trick77: <https://trick77.com/adding-ds3231-real-time-clock-raspberry-pi-3/>
- Lakubovskii, U. V., & Liajov, L. (1980). *Exploracion electrica* (tercera edicion ed.).
Barcelona: editorial Reverté S.A.
- Matt. (25 de mayo de 2015). *Adding a DS3231 Real Time Clock To The Raspberry Pi*.
Obtenido de raspberrypi-sp: <http://www.raspberrypi-spy.co.uk/2015/05/adding-a-ds3231-real-time-clock-to-the-raspberry-pi/>
- Miyara, F. (S.F.). *Conversores D/A y A/D*. Obtenido de facultad de ciencias exactas, ingenieria y agrimensura: <http://www.fceia.unr.edu.ar/enica3/da-ad.pdf>
- National instruments. (S.F.). *Fundamentos de acondicionamiento de señales para ingenieros*. Obtenido de ftp://ftp.ni.com/evaluation/signal_conditioning/23807_Engineer_s_guide_to_signal_conditioning_Spanish_localisation_HR.pdf
- Perez, M. (28 de Junio de 2013). *TUTORIAL RASPBERRY PI – 9. SERVIDOR FTP*.
Recuperado el 24 de noviembre de 2016, de geekytheory: <https://geekytheory.com/tutorial-raspberry-pi-9-servidor-ftp>
- Pipe. (10 de Febrero de 2015). *AYUDA: IP Fija en eth0 y en wlan0*. Recuperado el 20 de Noviembre de 2016, de raspberrypi: <https://www.raspberrypi.org/forums/viewtopic.php?f=76&t=33065>

- Reyes, C. (2006). *Microcontroladores PIC, programacion en basic*. Quito: RISPGRAF.
- Rosado, C. (2000). *Comunicacion por satelite: principios, tecnologias y sistemas*. Albadalejo, S.L.
- Ruiz Robredo, G. (2001). *electronica basica para ingenieros*. Madrid: Servicio de Reprografía facultad de ciencias universidad de Cantabria.
- Sabino, C. (1992). *El proceso de la investigacion*. Caracas: editorial Panapo.
- Vasquez, S. (17 de setiembre de 2013). *Raspberry Pi con Banda Ancha con modem ZTE MF180*. Recuperado el 20 de noviembre de 2016, de blogvazquezsaul: <http://blogvazquezsaul.blogspot.pe/2013/09/raspberry-pi-con-banda-ancha.html>
- Warner, T. L. (2014). *Hacking Raspberry PI*. indianapolis.





ANEXO 1: Código fuente en PYTHON del prototipo.

```
#####autor: Juan Albert Ramos Figueroa#####

#codigo realizado para mediciones de potencial espontaneo usan
Raspberry pi#

#! /usr/bin/python
import RPi.GPIO as GPIO #importa librerías para manejo de GPIO
import datetime          # importa librerías para manejo de fecha del
sistema
import time
import os
import Adafruit_ADS1x15 # librería del ADS1115

GPIO.setmode(GPIO.BCM)      #establece la numeración que se utilizara
para el GPIO
GPIO.setup(17, GPIO.OUT)    #PIN 17 como salida
GPIO.setwarnings(False)    # PIN 17 apagado

#####variables de
periodo#####
#con estas variables se puede modificar el periodo de muestreo desde
un segundo hasta varias horas
P_sec = 1
P_min = 0
P_hor = 0
#####
list_now = [] # variable tipo lista que almacena la hora actual
list_next = [] # variable tipo lista que almacena la hora de la
siguiente muestra
#####variables lista que almacenan
dia y mes #####
months = ["
", "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "S
etiembre", "Octubre", "Noviembre", "Diciembre"]
week day =
["Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"]

#####
path_main = "/home/pi/DATAPGR" # variable string que almacena la ruta
principal

path_potencial = path_main + "/" + "Potencial" # almacena la ruta
para medidas de potencia
path_status = path_main + "/" + "Estado" # almacena la ruta para
medida de la bateria

#####bloque que verifica si la dirección existe o
sera necesario crearla#####
if os.path.isdir(path_main) == 0:
    os.mkdir(path_main)
if os.path.isdir(path_potencial) == 0:
    os.mkdir(path_potencial)
if os.path.isdir(path_status) == 0:
    os.mkdir(path_status)
```

```
#####parámetros y métodos del ADC#####
adc = Adafruit_ADS1x15.ADS1115() # identificador del ADC
GAIN = 1 # variable de la ganancia
CHANNEL = 0 #variable del canal diferencial
CHANNEL_ST=2 # variable de canal normal
#####métodos del ADC #####
def dec_result(adc_value): # esta función toma el valor de la lectura
del #ADC que está en binario y la convierte a un valor en voltaje bajo
la #siguiente relación  $V = \frac{R}{2^{16}} * ADC$ , donde R = rango y ADC el valor de
#lectura.
    volt_value = (4.096/32768)*adc_value # y devuelve el valor en
voltios
    return volt_value

#####este método establece la hora en que se tomara la
siguiente muestra#####

def set_next_alarm(hour,minute,second):
    sample_second = second + P_sec
    if sample_second >59 :
        sample_second = sample_second-60
        minute = minute +1

    sample_minute = minute + P_min
    if sample_minute > 59:
        sample_minute = sample_minute - 60
        hour = hour + 1
    sample_hour = hour + P_hor
    if sample_hour > 23:
        sample_hour = sample_hour - 24
    return [sample_hour, sample_minute, sample_second ]
#####método que crea una nueva dirección segun pasa el
tiempo#####
#los datos vienen agrupados por días, es decir, cada día creara una
carpeta nueva
def path_define (path_const, year, month, day, weekday):
    path_year = path_const + "/" + str(year)
    path_month = path_const + "/" + str(year) + "/" + months[month]
    path_day = path_const + "/" + str(year) + "/" + months[month] +
"/" + str(day) + "-" + week_day[weekday]
    if (os.path.isdir(path_year) == 0):
        os.mkdir(path_year)
    if (os.path.isdir(path_month) == 0):
        os.mkdir(path_month)
    if (os.path.isdir(path_day) == 0):
        os.mkdir(path_day)
    return path_day
##### método que guarda el string que contiene los datos en la
carpeta correspondiente#####
def save_data(path, filename, in_string):
    file = open(path + "/" + filename, 'a')
    file.write(in_string)
    file.close()
#####obtiene la temperatura del procesador#####
def get_cpu_temp():
    tempFile = open( "/sys/class/thermal/thermal_zone0/temp" )
    cpu_temp = tempFile.read()
    tempFile.close()
    return float(cpu_temp)/1000
```

```
#####el código empieza desde acá #####

now = datetime.datetime.now() # obtiene la hora actual
list_now = [now.hour, now.minute, now.second] #guarda en una lista las
variables hora,minutos,segundos
list_next = set_next_alarm(now.hour,now.minute, now.second) # obtiene
la hora que siguiente donde se tomara una muestra

#print type (now.weekday())
#####primera condicional#####
#se permanecera en este bucle mientras le segundero no es igual 0.
while list_now[2] != 0: # compara segundero de la actual cada vez
    now = datetime.datetime.now() #obtiene nuevamente la hora actual
    list_now = [now.hour, now.minute, now.second] #guarda los datos en
una lista
    list_next = set_next_alarm(now.hour,now.minute, now.second)

#####acá empieza el bucle principal#####
while 1:
    now = datetime.datetime.now()
    list_now = [now.hour, now.minute, now.second]

    if list_now >= list_next: # compara constantemente que la hora
actual sea igual a la hora sigüiente establecida
        GPIO.output(17, True) #enciende el PIN 17
        list_next = set_next_alarm(now.hour,now.minute, now.second) #
determina la hora siguiente donde se tomara la muestra
        timestamp = now.strftime("%Y/%m/%d, %H:%M:%S") # guarda la
hora actual en un string

        pot_dif_volt = dec_result(adc.get_last_result()) #toma muestra
en la entrada diferencial
        adc.stop_adc() #detiene las
lecturas continuas en el canal diferencial
        sta_dif_volt = dec_result(adc.read_adc(CHANNEL_ST,GAIN)) #
#toma una muestra del canal ST
        sta_dif_volt = sta_dif_volt * 4.213 # para obtener el valor
del canal st en voltios
        print str(sta_dif_volt)
        val= dec_result(adc.start_adc_difference(CHANNEL,GAIN)) #inicia
muestreos indefinidos
        #la siguiente línea crea el nombre de archivo según hora cada
hora genera un nuevo archivo
        file_name = str(now.hour) + "POT" + str(now.year) +
months[now.month] + str(now.day) + ".csv"
        #esta línea crea un string con los datos del ADC y la fecha y
hora para ser guardado
        string_to_save = str(timestamp) + "," + str(pot_dif_volt) + ",V
,"+ "\n"
        #la variable path almacena la dirección de la carpeta donde se
guardaran los datos
        path = path_define(path_potencial, now.year, now.month,
now.day, now.weekday())
        #función que guarda los datos generados
        save_data (path, file_name, string_to_save)
        #genera un string del voltaje de batería y temperatura
        string_to_save_1 = str(timestamp) + "," + str(sta_dif_volt) + ",
V ," + str(get_cpu_temp()) + ", C" + "\n"
        print str(sta_dif_volt)+" " + str(val)
        #crea la dirección y guarda la información de batería y
```

```
temperatura de procesador  
    path_1 = path_define(path_status,now.year, now.month, now.day,  
now.weekday())  
    save_data (path_1, file_name, string_to_save_1)  
    GPIO.output(17, False)# apaga el led
```

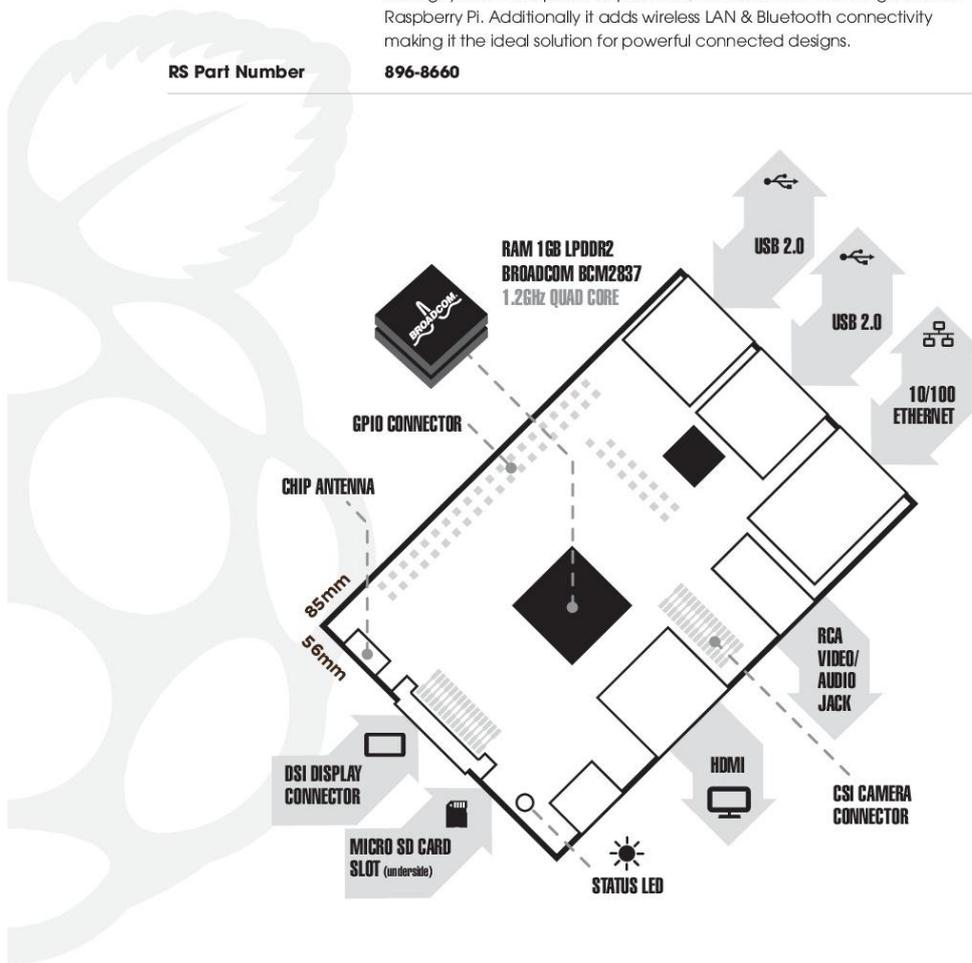


ANEXO 2: Características de Raspberry Pi 3 B+



Raspberry Pi 3 Model B

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



www.rs-components.com/raspberrypi

j



Raspberry Pi 3 Model B

Specifications

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

Connectors:

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

Key Benefits

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

Key Applications

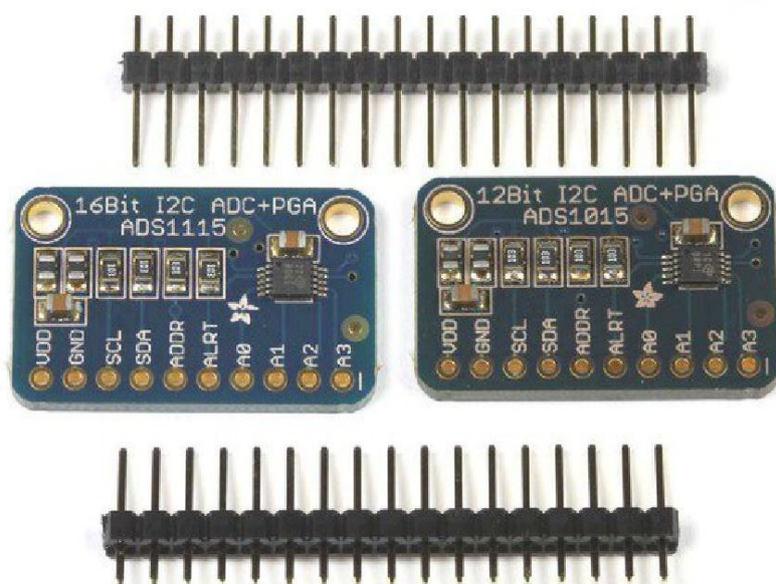
- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming



www.rs-components.com/raspberrypi

ANEXO 3: Características del módulo ads1115.

Overview



The ADS1115 and ADS1015 4-channel breakout boards are perfect for adding high-resolution analog to digital conversion to any microprocessor-based project. These boards can run with power and logic signals between 2v to 5v, so they are compatible with all common 3.3v and 5v processors. As many of 4 of these boards can be controlled from the same 2-wire I2C bus, giving you up to 16 single-ended or 8 differential channels. A programmable gain amplifier provides up to x16 gain for small signals.

These two boards are very similar, differing only in resolution and speed. The ADS1115 has higher resolution and the ADS1015 has a higher sample rate.

ADS1115 Features:

- Resolution: 16 Bits
- Programmable Sample Rate: 8 to 860 Samples/Second

- Power Supply/Logic Levels: 2.0V to 5.5V
- Low Current Consumption: Continuous Mode: Only 150 μ A Single-Shot Mode: Auto Shut-Down
- Internal Low-Drift Voltage Reference
- Internal Oscillator
- Internal PGA: up to x16
- I2C Interface: 4-Pin-Selectable Addresses
- Four Single-Ended or 2 Differential Inputs
- Programmable Comparator

ADS1015 Features:

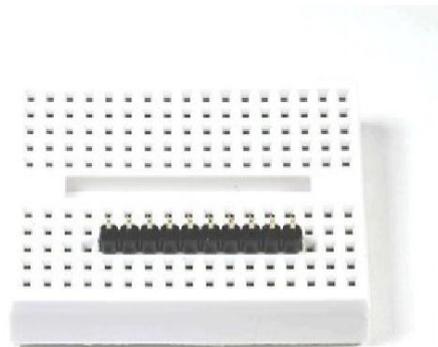
- Resolution: 12 Bits
- Programmable Sample Rate: 128 to 3300 Samples/Second
- Power Supply/Logic Levels: 2.0V to 5.5V
- Low Current Consumption: Continuous Mode: Only 150 μ A Single-Shot Mode: Auto Shut-Down
- Internal Low-Drift Voltage Reference
- Internal Oscillator
- Internal PGA: up to x16
- I2C Interface: 4-Pin-Selectable Addresses
- Four Single-Ended or 2 Differential Inputs
- Programmable Comparator



Assembly and Wiring

Assembly:

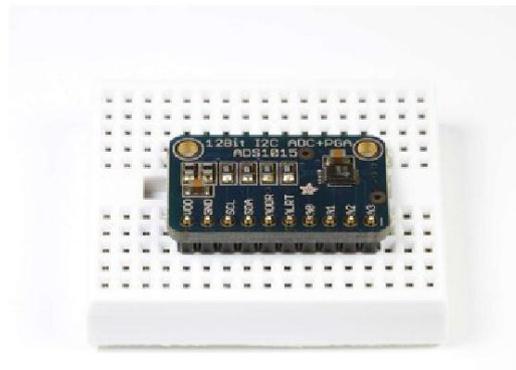
The board comes with all surface-mount parts pre-soldered. For breadboard use, the included header-strip should be soldered on:



Prepare the header strip

Cut the supplied header strip to length and insert it long-pins-down in your breadboard to hold it for soldering.

•



Position the breakout board

Place the breakout board on the header pins.

•



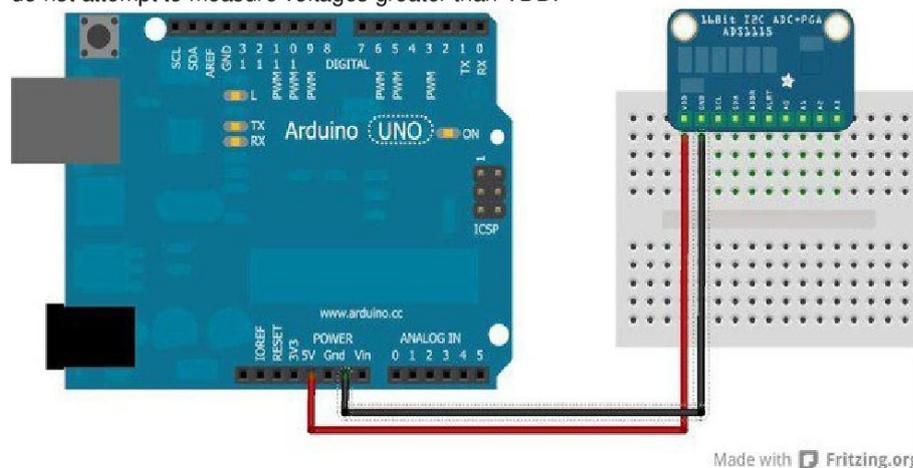
Solder!

Solder each pin for a good electrical connection.

Wiring:

Power

First connect VDD and GND. These boards will work with either a 3.3v or a 5v supply. The diagram below shows connection to the Arduino 5v pin. The absolute maximum analog input voltage is $VDD + 0.3v$. To avoid damage to the chip, do not attempt to measure voltages greater than VDD.

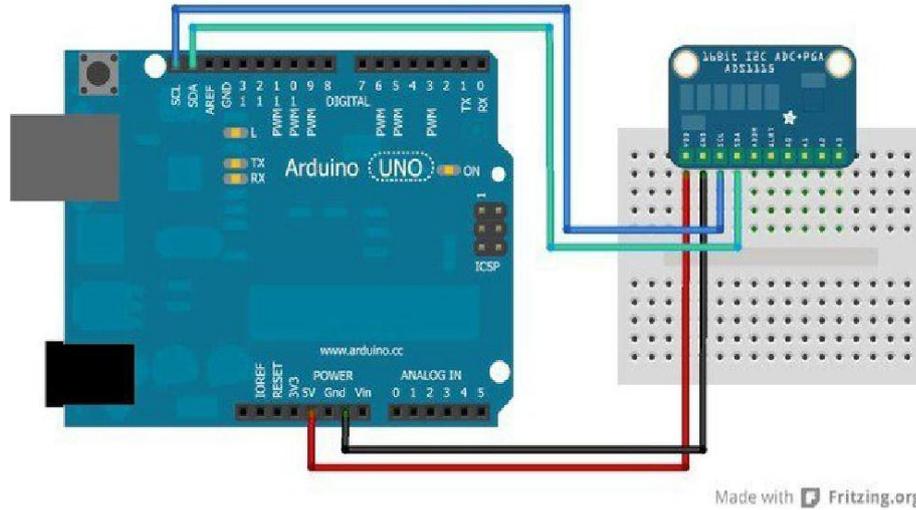


Made with Fritzing.org

I2C Connections

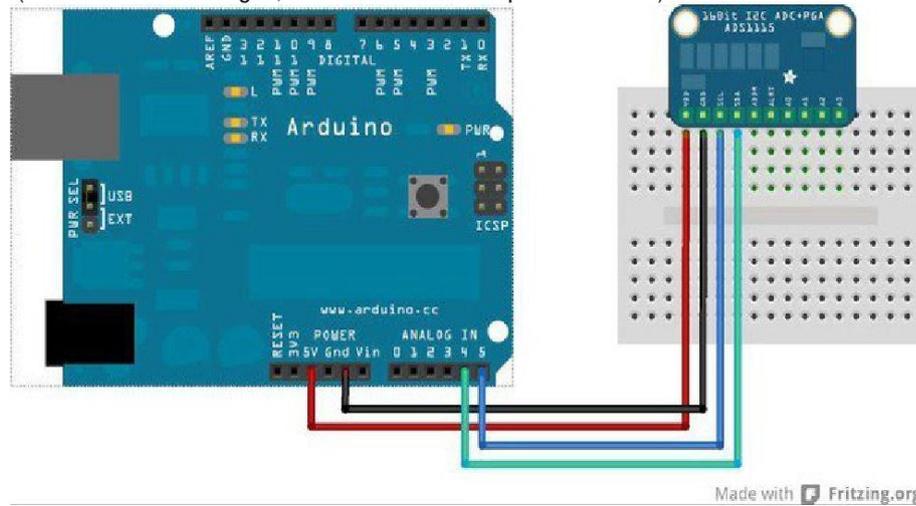
I2C requires just 2 pins to communicate. These can be shared with other I2C devices. For

R3 and later Arduinos (including MEGA and DUE models), connect SDA->SDA and SCL->SCL.



I2C "Classic"

For older Arduino boards without dedicated SDA and SCL pins, connect as shown below. (For older Arduino Megas, SDA and SCL are on pins 20 and 21)

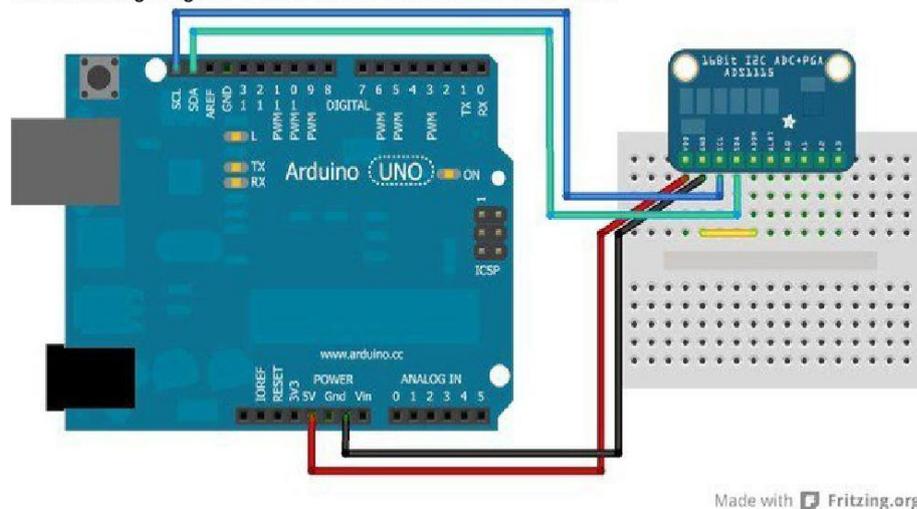


I2C Addressing

The ADS11x5 chips have a base 7-bit I2C address of 0x48 (1001000) and a clever addressing scheme that allows four different addresses using just one address pin (named **ADR** for ADdRess). To program the address, connect the address pin as follows:

- 0x48 (1001000) ADR -> GND
- 0x49 (1001001) ADR -> VDD
- 0x4A (1001010) ADR -> SDA
- 0x4B (1001011) ADR -> SCL

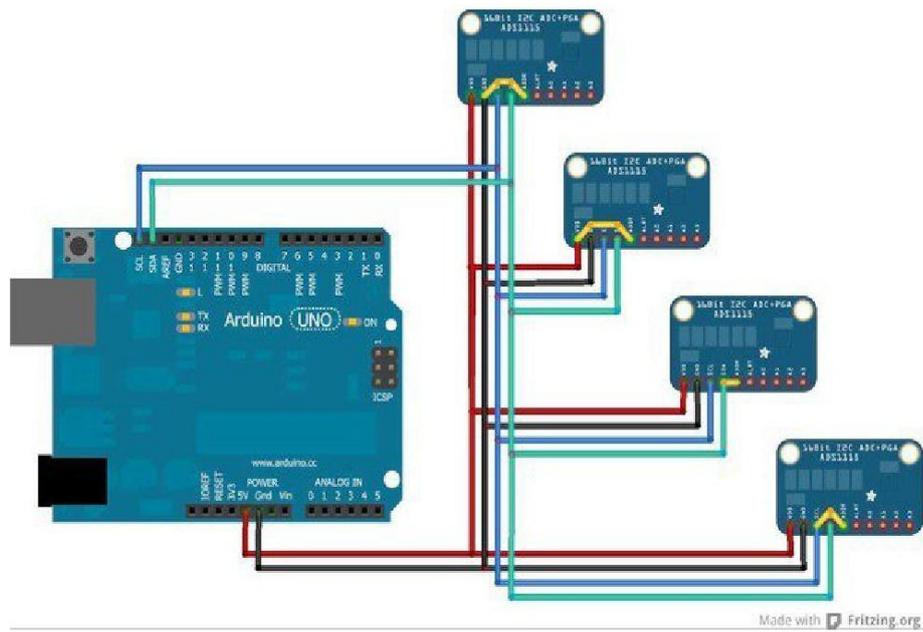
The following diagram shows one board addressed as 0x48:



Made with Fritzing.org

Multiple Boards

By assigning each board a different address, up to 4 boards can be connected as below:



ANEXO 4: Características técnicas de cajas ID.

Sección catálogo: 1.3

Serie SCABOX

> CAJA DE DERIVACIÓN MURAL



> VERSIONES

-  Con conos pasacables - IP44
- Tapa baja montada a presión
-  Con conos pasacables - IP55
- Tapa baja montada con tornillos
-  Sin conos - IP56
- Tapa baja montada con tornillos
-  Sin conos - IP56
- Tapa alta montada con tornillos

> NORMAS DE REFERENCIA

EN 60670-1
Envoltorios para aparatos para instalaciones eléctricas fijas para uso doméstico y similar.
Parte 1: Prescripciones generales.

EN 60670-22
Envoltorios para aparatos para instalaciones eléctricas fijas para uso doméstico y similar.
Parte 22: Prescripciones para cajas de derivación.

> MARCA DE CALIDAD

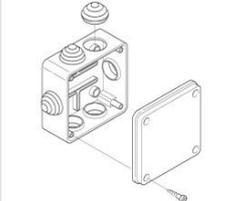


> CARACTERÍSTICAS TÉCNICAS

Grado de protección:	IP44 - IP55 - IP56
Temperatura de uso según normas de referencia:	-25°C/+35°C
Máxima temperatura de funcionamiento:	60°C
Prueba de hilo incandescente:	650°C
Material:	Tecopolimero
Grado IK a 20°:	IK04 - IK08
Aislamiento total III:	SI
Libre de halógenos:	SI (Itas)
Fijación guía DIN:	SI
Color:	Gris RAL 7035

> RESISTENCIA A LOS AGENTES QUÍMICOS Y ATMOSFÉRICOS

Solución salina	Ácidos		Bases		Disolventes			Aceite Mineral	Rayos UV
	Concentradas	Diluidos	Concentradas	Diluidos	Hexano	Benzol	Acetona		
Alta	Baja	Alta	Baja	Alta	Nula	Nula	Nula	Baja	Alta



Caja de derivación mural 100x100mm. Ejecución con caja baja (50mm) con conos pasacablesIP55. Cierre de tapa con tornillos.

> EJEMPLOS DE APLICACIÓN



Serie SCABOX

636

637

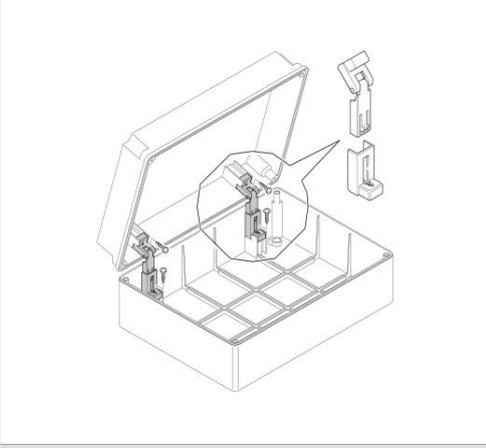
Sección catálogo: 1.3

Serie SCABOX

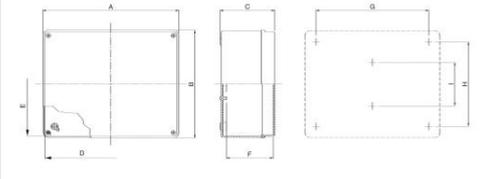
> KIT BISAGRAS

Interesante novedad para la Serie SCABOX. Está disponible un práctico kit de bisagras que permite abrir la tapa a libro y fijarla a la base.
El kit se puede aplicar de forma opcional dentro del contenedor de forma simple y rápida; las SCABOX adquieren mayores prestaciones y son perfectas para la fase de instalación; durante las operaciones de cableado y son sumamente prácticas para las posteriores fases de mantenimiento e inspección, porque mantienen la tapa impermeable y articulada a la base.

Serie SCABOX Kit bisagras (3 unidades) 654.0085



> DIMENSIONES



Dimensiones internas			Dimensiones externas			Montaje en pared			
D	E	F	A	B	C	G	H	I	
mm	mm	mm	mm	mm	mm	mm	mm	mm	
600.001	685	685	35	676	673	41	/	/	32
600.002	685	685	40	667	667	46	/	/	46
600.003	80	80	40	67	67	46	/	/	46
600.004	100	100	50	108	108	58	64	64	/
600.005	120	120	50	128	128	58	84	44	/
600.006	150	150	50	158	158	78	116	74	/
600.007	190	140	50	188	148	80	156	106	/
600.008	240	190	80	248	198	100	206	156	/
600.009	300	220	100	308	228	120	266	186	/
600.010	360	280	120	368	288	140	346	266	124
600.011	450	370	130	460	380	150	406	296	180
600.204	100	100	50	108	108	58	64	64	/
600.205	120	120	50	128	128	58	84	44	/
600.206	150	150	50	158	158	78	116	74	/
600.207	190	140	50	188	148	80	156	106	/
600.208	240	190	80	248	198	100	206	156	/
600.209	300	220	100	308	228	120	266	186	/
600.210	360	280	120	368	288	140	346	266	124
600.211	450	370	130	460	380	150	406	296	180
600.404	100	100	80	108	108	88	64	64	/
600.406	240	190	120	248	198	120	206	156	/
600.409	300	220	150	308	228	150	266	186	/
600.410	360	300	170	368	308	170	346	266	124
600.411	450	370	180	460	380	180	406	296	180

(Dimensiones en mm)

638

639

168

Repositorio Institucional UNA-PUNO

No olvide citar esta tesis