



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA
ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON
ACCESO A LA APN DE INSIVUMEH**

Carlos Alejandro Montenegro Ruano
Asesorado por el Ing. Kenneth Issur Estrada Ruiz

Guatemala, enero de 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA
ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON
ACCESO A LA APN DE INSIVUMEH**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CARLOS ALEJANDRO MONTENEGRO RUANO
ASESORADO POR EL ING. KENNETH ISSUR ESTRADA RUIZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, ENERO DE 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
EXAMINADOR	Ing. Kenneth Issur Estrada Ruiz
EXAMINADOR	Ing. Natanael Jonathan Requena Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 03 de febrero de 2020.

Carlos Alejandro Montenegro Ruano

Universidad de San Carlos de
Guatemala



Guatemala, 31 de octubre de 2021.
Ref.EPS.DOC.455.10.2021.

Ing. Oscar Argueta Hernández
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Argueta Hernández.

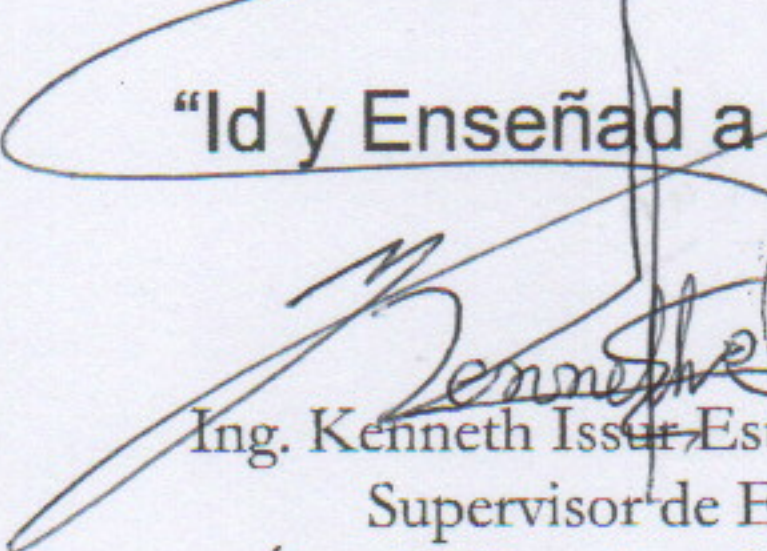
Por este medio atentamente le informo que como Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Carlos Alejandro Montenegro Ruano** de la Carrera de Ingeniería Electrónica, Registro Académico No. **201212530 y CUI 2322 66131 0101**, procedí a revisar el informe final, cuyo título es **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH”**.

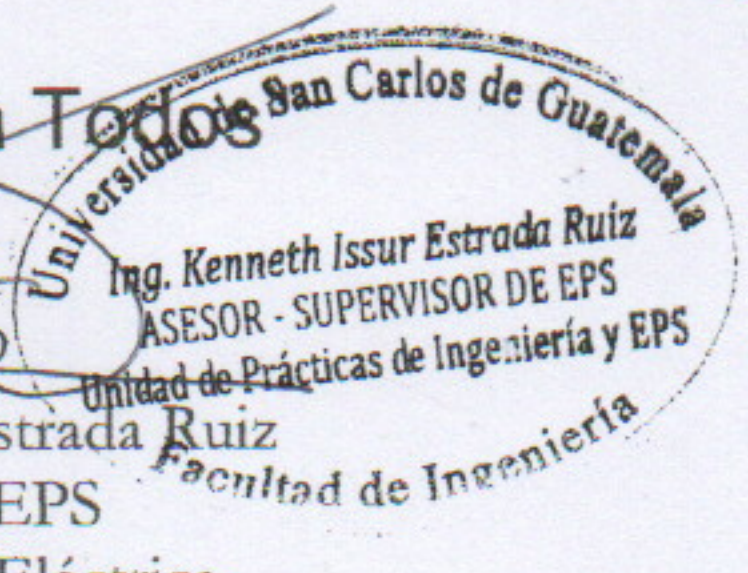
En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”


Ing. Kenneth Issur Estrada Ruiz
Supervisor de EPS
Área de Ingeniería Eléctrica



c.c. Archivo
KIER/ra

Universidad de San Carlos de
Guatemala



Facultad de Ingeniería
Unidad de EPS

Guatemala 31 de octubre de 2021.
Ref.EPS.D.246.10.2021.

Ing. Armando Alonso Rivera Carrillo
Director Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Presente

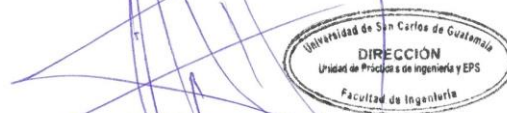
Estimado Ingeniero Rivera Carrillo.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH"** que fue desarrollado por el estudiante universitario, **Carlos Alejandro Montenegro Ruano**, quien fue debidamente asesorado y supervisado por el Ing. Kenneth Issur Estrada Ruiz.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor -Supervisor de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"



Ing. Oscar Argueta Hernández
Director Unidad de EPS

/ra



REF. EIME 162 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; CARLOS ALEJANDRO MONTENEGRO RUANO titulado: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH, procede a la autorización del mismo.

Ing. Armando Alonso Rivera Carrillo



GUATEMALA, 16 DE NOVIEMBRE 2021.

Facultad de Ingeniería

Decanato
24189101-
24189102
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.047.2022

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH**, presentado por: **Carlos Alejandro Montenegro Ruano**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Aurelia Anabela Cordova Estrada

Decana

Guatemala, enero de 2022

AACE/gaoc

ACTO QUE DEDICO A:

- Dios** Por ser una importante influencia en mi carrera y darme la oportunidad de vivir.
- Mi madre** Luz María Montenegro Ruano por su cariño, apoyo, inspiración y soporte durante mi vida, este triunfo también es de ella.
- Mis abuelos** Nidia Edith Ruano Reyes de Montenegro (q. e. p. d.) y Luis Carlos René Montenegro Monterroso (q. e. p. d.), un recuerdo inolvidable y por ser una inspiración en mi vida.
- Amigos** Por formar parte importante en esta aventura.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser el Alma Mater en mis estudios universitarios.
Facultad de Ingeniería	Por ser el instrumento de enseñanza durante mi carrera universitaria.
INSIVUMEH	Por ser la institución que abrió sus puertas y permitió realizar este proyecto.
Ing. Kenneth Estrada	Por ser un apoyo en todo momento y guiarme durante la finalización de este proceso.
Lic. Willson García	Por ser un profesional ejemplar y por su apoyo incondicional durante el desarrollo del presente proyecto.
Mi madre	Por ser el pilar principal en el transcurso de mi carrera.
Mis amigos	Por hacer una experiencia inolvidable. Por compartir los buenos y malos momentos.
Colegio Salesiano Don Bosco	Por ser parte de mi formación durante una de las mejores épocas de mi vida.

A mis catedráticos

Por guiarme en el transcurso de esta gran
aventura, por su paciencia, bondad y sabiduría.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS	XV
GLOSARIO	XVII
RESUMEN.....	XXIII
OBJETIVOS.....	XXV
INTRODUCCIÓN	XXVII
1. GENERALIDADES DEL INSTITUTO NACIONAL DE SISMOLOGÍA, VULCANOLOGÍA, METEOROLOGÍA E HIDROLOGÍA	1
1.1. Historia	1
1.2. Misión	2
1.3. Visión.....	3
1.4. Estructura de la institución.....	3
1.4.1. Departamento Administrativo - Financiero.....	3
1.4.2. Departamento de Investigación y Servicios Hídricos	4
1.4.3. Departamento de Investigación y Servicios Meteorológicos	5
1.4.4. Departamento de Investigación y Servicios Geofísicos.....	5
1.5. Organigrama.....	6
2. TELEMETRÍA POR GPRS DE UNA ESTACIÓN METEOROLÓGICA.....	7
2.1. Estación meteorológica	7
2.1.1. Variables meteorológicas	8

	2.1.1.1.	Temperatura.....	8
	2.1.1.2.	Presión atmosférica.....	11
	2.1.1.3.	Humedad.....	13
	2.1.1.4.	Precipitación.....	15
	2.1.1.5.	Viento.....	17
	2.1.1.6.	Radiación solar.....	21
	2.1.1.7.	Telemetría.....	25
	2.1.2.	Funcionamiento.....	26
2.2.	GPRS.....		27
	2.2.1.	Módulo GPRS/GSM SIM900.....	28
	2.2.2.	Comandos AT.....	30
	2.2.3.	APN.....	31
	2.2.3.1.	Tipos de APN.....	32
	2.2.4.	Microcontroladores.....	32
	2.2.4.1.	Historia.....	33
	2.2.4.2.	Raspberry Pi.....	34
	2.2.4.3.	Raspberry Pi 3 B+.....	35
	2.2.4.4.	Software operativo de la Raspberry Pi	35
	2.2.4.5.	Puertos GPIO de Raspberry Pi modelo 3 B+.....	37
	2.2.4.6.	Arduino.....	40
	2.2.4.7.	Arduino NANO.....	41
	2.2.4.8.	Puertos GPIO Arduino NANO.....	43
3.	SENSORES QUE COMPONEN LA ESTACIÓN METEOROLÓGIA DE BAJO COSTO IMPRESO EN 3D.....		45
	3.1.	Sensor de presión BMP280.....	45
	3.2.	Sensor de temperatura MCP9808.....	47

3.3.	Sensor de humedad HTU21D	48
3.4.	Sensor de intensidad lumínica SI1145	55
4.	DISEÑO DE UNA ESTACIÓN METEOROLÓGICA IMPRESA EN 3D... 61	
4.1.1.	Conjunto del protector de radiación para los sensores BMP280, MCP9808 y HTU21D.....	63
4.1.2.	Conjunto de acople para recolección de datos y protección de sensor SS541A, sensor de velocidad de viento	67
4.1.3.	Conjunto de acople para recolección de datos y protección del sensor Littelfuse 55300-00-02-A.....	69
4.1.4.	Conjunto de acople para recolección de datos y protección del sensor Honeywell SS451A, sensor de precipitación.....	70
4.1.5.	Conjunto de protección para el sensor de intensidad lumínica SI1145.....	74
4.2.	Convertor ADC	76
4.2.1.	MCP3002.....	76
4.3.	Ensamblaje de sensores a <i>Raspberry Pi</i>	80
4.4.	Comunicación GPRS.....	82
5.	IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH	85
5.1.	Sistema operativo <i>Raspberry Pi</i>	85
5.2.	Configuración <i>Raspberry Pi</i>	94
5.2.1.	Código Pi de Adafruit.....	105
5.2.2.	Prueba I2C.....	106

5.2.3.	Prueba SPI	107
5.2.4.	Instalación de Python	108
5.3.	Programación de sensores	109
5.3.1.	Sensor de presión BMP280.....	109
5.3.2.	Sensor de humedad HTU21D	111
5.3.3.	Sensor de temperatura MCP9808.....	114
5.3.4.	Sensor Littelfuse 55300-00-002-A.....	115
5.3.5.	Sensor SS451A.....	119
5.3.5.1.	Velocidad de viento	119
5.3.5.2.	Precipitación.....	121
5.3.6.	Sensor de intensidad lumínica SI1145	122
5.3.6.1.	Configuración de SIM.....	124
5.3.6.2.	Configuración de GPRS en Raspberry Pi.....	128
5.3.7.	Base de datos	134
5.3.8.	Acciones automáticas por tiempos.....	142
5.4.	Incorporación sensor de calidad de Aire a la estructura inicial	145
5.5.	Manual de usuario.....	153
5.5.1.	Instalación de una estación meteorológica.....	154
5.5.1.1.	Ubicación de la estación.....	154
5.5.1.2.	Infraestructura de la estación	155
5.5.1.3.	Sistema de protección contra descargas electro-atmosféricas.....	155
5.6.	Manual de mantenimiento	156
5.6.1.	Hardware.....	156
5.6.2.	Software	157
	CONCLUSIONES.....	161

RECOMENDACIONES.....	163
BIBLIOGRAFÍA.....	165
APÉNDICES	167

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Organigrama de INSIVUMEH	6
2.	Ley de distribución de Maxwell	9
3.	Pantalla Stevenson	10
4.	Barómetro	12
5.	Psicrómetro digital.....	14
6.	Proceso de crecimiento de una gota en el interior de una nube	16
7.	Pluviómetro	17
8.	Escala de <i>Beaufort</i>	18
9.	Anemómetro.....	19
10.	Veleta	20
11.	Rosa de vientos	21
12.	Pronóstico de índice ultravioleta	23
13.	Espectro electromagnético.....	24
14.	Piranómetro.....	25
15.	Telemetría	27
16.	Módulo GPRS/GSM SIM900.....	29
17.	TMS 1 000.....	34
18.	Pines GPIO <i>Raspberry Pi</i> modelo 3 B+	39
19.	Ejemplo comando <i>gpio readall</i>	40
20.	Microcontrolador ATMEGA328P	42
21.	Sensor de presión BMP280	46
22.	Sensor MCP9808.....	48
23.	Sensor de humedad HTU21D	49

24.	Actuador magnético de rotación	50
25.	Actuador magnético	51
26.	Sistema de numeración de piezas sensores <i>Littelfuse</i>	51
27.	Señal de salida	52
28.	Sensor 55 300-00-02-A.....	53
29.	Sensor SS451A	55
30.	Diagrama de sincronización de bits I2C.....	56
31.	Diagrama de comunicación I2C.....	57
32.	Diagrama de secuencia de inicio protocolo I2C	58
33.	Diagrama de secuencia de finalización protocolo I2C	59
34.	Impresora 3D Afinia H480.....	62
35.	Impresora 3D Afinia H800.....	63
36.	Esquema de conexión sensores BMP280, MCP9808 y HTU21D.....	65
37.	Vista 3D conectores comunes	66
38.	Impresión 3D conector común base y bloque de pines	66
39.	Impresión 3D conector común base y tapa.....	67
40.	Cazoleta y equipo para anemómetro	68
41.	Diagrama de conexión sensor SS451.....	69
42.	Veleta y sensor <i>Littelfuse</i> SS300-00-02-A	70
43.	Diagrama de conexión sensor SS451.....	71
44.	Topes desgastados para calibración de toma de datos sobre precipitación.....	72
45.	Cubeta o balancín para toma de muestras de precipitación	73
46.	Cono superior y embudo.....	73
47.	Sensor y soporte para sensor SI1145.....	75
48.	Fijación de PTFE a tapa	75
49.	<i>Raspberry Pi</i> ADC, MCP3002.....	76
50.	Conexión de los pines SPI.....	78
51.	Diagrama del circuito del MCP2003.....	78

52.	Diseño en 3D de la cubierta del circuito integrado MCP2003	79
53.	Esquema de conexión sensores, circuito integrado y <i>Raspberry Pi</i>	80
54.	Diseño 3D de conector para GPIO <i>Raspberry Pi</i>	81
55.	Diagrama de comunicación GPRS.....	82
56.	Sitio web oficial <i>Raspberry Pi</i> para descarga de <i>software</i>	86
57.	Botón para ver todas las opciones de descarga	87
58.	Descarga imagen de <i>Raspberry Pi OS</i>	88
59.	Descomprimir archivo.zip	88
60.	Micro SD y adaptador.....	89
61.	Aplicación SD Card Formatter.....	89
62.	Hardware reconocido por el ordenador	90
63.	Mensaje de advertencia al formatear microSD	91
64.	Win32 Disk Imager	92
65.	Agregar archivo .img a Win32 Disk Imager	92
66.	Mensaje de advertencia al escribir archivo .img en microSD	93
67.	Proceso de escritura archivo .img en micro SD	93
68.	Finalización de grabado archivo .img en microSD	94
69.	Interacción con la interfaz gráfica de <i>Raspberry Pi</i>	95
70.	Interface Options de <i>Raspberry Pi</i>	96
71.	Utilización <i>Advanced IP Scanner</i>	97
72.	Comunicación SSH desde cmd.....	98
73.	Programa Putty	99
74.	Alerta Putty.....	99
75.	Ingreso por ssh utilizando Putty	100
76.	Fichero /etc/hosts	101
77.	Cambio de nombre en fichero hosts.....	102
78.	Fichero /etc/hostname	102
79.	Cambio de nombre en fichero hostname	103
80.	Interfaces de <i>Raspberry Pi</i>	105

81.	Pip3 <i>install adafruit flash</i>	106
82.	Comando sudo <i>i2cdetect -y 0</i>	107
83.	Comando <i>ls -l /dev/spidev*</i>	108
84.	Código sensor BMP280 parte 1	110
85.	Código sensor BMP280 parte 2	111
86.	Código sensor BMP280 parte 3	111
87.	Código sensor HTU21D parte 1	112
88.	Código sensor HTU21D parte 2	113
89.	Código sensor HTU21D parte 3	113
90.	Código sensor HTU21D parte 4	114
91.	Código sensor MCP9808	115
92.	Código sensor <i>Littelfuse</i> 55300-00-002-A parte 1	116
93.	Código sensor <i>Littelfuse</i> 55300-00-002-A parte 2	117
94.	Código sensor <i>Littelfuse</i> 55300-00-002-A parte 3	118
95.	Código sensor <i>Littelfuse</i> 55300-00-002-A parte 4	118
96.	Código velocidad de viento sensor SS451A parte 1	119
97.	Código velocidad de viento sensor SS451A parte 2	120
98.	Código velocidad de viento sensor SS451A parte 3	120
99.	Código precipitación sensor SS451A parte 1	121
100.	Código precipitación sensor SS451A parte 2	122
101.	Sensor de intensidad lumínica SI1145	123
102.	Implementación de GPRS a estación meteorológica	123
103.	Página de inicio configuración APN	124
104.	Página inicial de configuraciones, APN	125
105.	Administración de perfiles BROADBAND TIGO GT	125
106.	Creación de nuevo perfil	126
107.	Aplicar nuevo perfil creado	127
108.	Búsqueda de información de dispositivo	127
109.	Información de dispositivo	128

110.	Conexión USB – TTL a GRPS	129
111.	Directorio peers	130
112.	Creación archivo rnet	131
113.	Archivo gprs	132
114.	Ifconfig pp0.....	133
115.	Pantalla de configuración XAMPP	136
116.	Pantalla de inicio phpMyAdmin	136
117.	Creación de tabla datos	137
118.	Contenido tabla datos	138
119.	Conexión a base de datos.....	139
120.	Presentación de conexión al inicio de programa.....	140
121.	Insertar datos en tabla.....	140
122.	Insertar datos en tabla de velocidad de viento	140
123.	Resultado base de datos.....	141
124.	Comparativo IR y UV.....	142
125.	Comando sudo crontab -e.....	144
126.	<i>Script(s)</i> a ejecutar	145
127.	Estructura inicial estación meteorológica impresa en 3D	146
128.	Incorporación trípode	147
129.	Implementación caja NEMA 4	148
130.	Prensaestopas PG-9	148
131.	Unidad de alimentación.....	151
132.	Inclinación de panel solar.....	151
133.	Resultado final de estación meteorológica impresa en 3D y acople de sensor de calidad de aire	152
134.	Entrega estación meteorológica a personal de INSIVUMEH	153
135.	Fichero de <i>script</i>	158
136.	Programación de mantenimiento propuesto.....	159

TABLAS

I.	Comandos AT principales.....	31
II.	Especificaciones técnicas <i>Raspberry Pi</i> 3 Modelo B+	36
III.	Características ATMEGA328P.....	42
IV.	Características GPIO Arduino NANO	43
V.	Especificaciones técnicas sensor BMP280.....	45
VI.	Descripción de pines BMP280.....	46
VII.	Especificaciones técnicas sensor MCP9808	47
VIII.	Descripción pines MCP9808.....	47
IX.	Descripción de pines HTU21D.....	49
X.	Opciones de terminación sensores <i>Littelfuse</i>	52
XI.	Lista de piezas para protector de radiación para sensores BMP280, MCP9808 y HTU21D	64
XII.	Piezas conector común.....	65
XIII.	Lista de piezas utilizadas en anemómetro	68
XIV.	Lista de piezas utilizadas en anemómetro	69
XV.	Lista de piezas utilizadas en el pluviómetro.....	71
XVI.	Listas de piezas utilizadas en el sensor de radiación solar.....	74
XVII.	Conexión de pines SPI	77
XVIII.	Piezas de ensamble circuito integrado MCP2003	79
XIX.	Posición de GPIO <i>Raspberry Pi</i>	81
XX.	Longitudes de cable utilizados para la estación meteorológica impresa en 3D.....	82
XXI.	Materiales conexión GPRS a <i>Raspberry Pi</i>	128
XXII.	Consumo energético.....	149
XXIII.	Sistema de alimentación.....	149
XXIV.	Cálculo de autonomía	150

XXV.	Propuesta de cronograma para mantenimiento	158
XXVI.	Formato de programación de mantenimiento.....	160

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
b	bit
B	Byte
DC	Corriente continua
°C	Grados Celsius
°F	Grados Fahrenheit
K	Grados Kelvin
°	Grados sexagesimales
hPa	Hectopascales
Hz	Hertz
HR	Humedad relativa
KHz	Kilo-Hertz
m/s	Metro por segundo
mA	Miliamperio
mm	Milímetro
ms	Milisegundo
Km/h	Ordenados alfabéticamente; siglas en glosario
inHg	Pulgadas de mercurio
T	Temperatura
V	Volt
Vcc	Voltaje de alimentación
Vin	Voltaje de entrada
Vo	Voltaje de salida

GLOSARIO

ADC	Conversor análogo a digital, permite procesar señales analógicas de manera digital para facilitar su procesamiento.
Apache	Servidor web de distribución libre y código abierto.
ARM	Arquitectura RISC, ordenador de conjunto reducido de instrucciones
ASA	Terpolímero amorfo termoplástico, constituido por acrilonitrilo, con excepcional resistencia a la intemperie.
AT	Instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem.
Bit	Señal electrónica que puede estar encendida (1) o apagada (0).
Byte	Unidad fundamental del dato, un byte son ocho bits contiguos.
Climatología	Estudia la variación de los elementos meteorológicos cerca de la tierra en un área reducida.

FFGS	Sistema de guía de inundaciones repentinas con cobertura global. (Flash Flood Guidance System with Global Coverage, en inglés).
GND	Común de la interconexión entre todos los dispositivos.
GPIO	Puerto de interconexiones físicas de la <i>Raspberry pi</i> , digitales, los cuales pueden ser configurados como entradas o salidas (<i>general prupose input/output</i> , en inglés).
GPRS	Paquete general de servicios vía radio (<i>General Packet Radio Service</i> , en inglés).
GSM	Sistema global de comunicaciones móviles (<i>General System for Mobile Communication</i> , en inglés).
Hardware	Parte física tangible de un sistema informático.
HDMI	Interfaz multimedia de alta definición (<i>high definition multimedia interface</i> , en inglés).
I2C	Bus de comunicación digital síncrono, multimaestro, multiesclavo, con comunicación de paquetes de un solo extremo. (<i>inter-integrates circuit</i> , en inglés).
INSIVUMEH	Instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología.

IR	Radiación electromagnética, de mayor longitud de onda que la luz visible, pero de menor longitud que las de microondas.
LoRa	Tecnología inalámbrica que utiliza un tipo de modulación en radiofrecuencia patentado por Semtech.
Meteorología	Ciencia encargada del estudio de las diferentes propiedades, eventos y fenómenos que tienen lugar en la atmósfera de la tierra.
microSD	Formato para tarjetas de memoria <i>flash</i> (<i>Micro Secure Digital</i> , en inglés).
Módem	Dispositivo que convierte señales digitales en analógicas, o viceversa, para poder ser transmitidas a través de líneas de teléfono, cable coaxial, fibra óptica y microondas.
MySQL	Gestor de bases de datos que permite la interacción diferentes lenguajes de programación.
NWS IAO	Oficina de asuntos internacionales del servicio meteorológico nacional (National Weather Service International Activities Office).
OS	Sistema operativo.

PHP	Lenguaje de código abierto adecuado para el desarrollo web (Hypertext Preprocessor, en inglés).
PM10	Partículas sólidas o líquidas de polvo, ceniza, hollín, partículas metálicas, cemento o polen dispersas en la atmósfera, cuyo diámetro es menor que 10 micrómetro.
PM2,5	Partículas muy pequeñas suspendidas en el aire que tiene un diámetro de menos de 2,5 micras.
Putty	Cliente SSH y Telnet con que podemos conectarnos a servidores remotos iniciando una sesión, el cual permite ejecutar comandos.
PVC	Polímero formado de macromoléculas denominadas monómeros, sus siglas corresponden a Policloruro de Vinilo.
<i>Python</i>	Lenguaje de programación de código legible.
<i>Roaming</i>	Capacidad de cambiar de un área de cobertura celular a otra sin interrupción en el servicio o pérdida de conectividad.
RPi	Microcomputador de placa reducida, de bajo costo, conocida como <i>Raspberry Pi</i> .

SCL	(System Clock) es la línea de pulsos del reloj que sincroniza el sistema.
SDA	(System Data) es la línea por la que se mueven los datos entre los dispositivos.
SIM	Tarjeta inteligente desmontable usada en teléfonos móviles y módems (<i>Suscriber Identity Module</i> , en inglés).
SoC	Sistema integrado con todos los módulos que componen un computador o cualquier otro sistema informático o electrónico (<i>system on chip</i> , en inglés)
Software	Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
SSH	Protocolo que permite el acceso remoto a un servidor por medio de un canal seguro, en el que toda la información es cifrada (<i>Secure Shell</i> , en inglés).
TCP/IP	Es una descripción de protocolos de red desarrollada en la década de 1970.
UCAR	Corporación Universitaria para la Investigación Atmosférica (University Corporation for Atmospheric Research, en inglés).

USB	Responde a las siglas <i>Universal Serial Bus</i> , que se traduce como puerto de seriado universal, y permite enlazar diversos periféricos a un dispositivo electrónico.
UV	Abreviatura de radiación ultravioleta.
VPN	Es una conexión segura y cifrada entre dos redes, o entre un usuario determinado y una red. (<i>Virtual Private Network</i> , en inglés).
WMO	Organización Meteorológica Mundial (World Meteorological Organization, en inglés).
XAMPP	Distribución de Apache que incluye varios <i>softwares</i> libres.

RESUMEN

Las observaciones meteorológicas cuantificadas son importantes en la agricultura, prevención de riesgos y generación de información para estudios posteriores. Uno de los instrumentos más utilizados son las estaciones meteorológicas automatizadas. Las estaciones que se utilizan en Guatemala son generalmente tecnologías adquiridas en países desarrollados, por lo que los equipos que sufren desperfectos no pueden ser reparados por costos.

El presente trabajo presenta el desarrollo de un prototipo de estación meteorológica automatizada, que permite incorporar distintos sensores y tiempos de muestreo, según las necesidades. Con el objetivo principal de reducir costos y generar información en tiempo real, el cual permitirá realizar su instalación en cualquiera de las regiones del país, y ser monitoreado continuamente desde las oficinas centrales del Instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología, INSIVUMEH.

El sistema implementado cuenta con la opción de análisis histórico de datos y está diseñado con una capacidad de procesamiento de datos cada minuto, manteniendo su base de datos en tiempo real.

OBJETIVOS

General

Diseñar e implementar un sistema de telemetría aplicado a una estación meteorológica de bajo costo, impreso en 3D.

Específicos

1. Implementar el uso de GPRS a una estación meteorológica, para transmisión de datos.
2. Implementar un código de programación para adquirir, guardar y gestionar datos suministrados por los distintos sensores.
3. Implementar el uso de Access Point Name (APN) en una estación meteorológica de bajo costo, impreso en 3D.
4. Establecer la transferencia y recepción de parámetros meteorológicos desde la estación meteorológica de bajo costo, impreso en 3D, hacia el servidor asignado de INSIVUMEH.

INTRODUCCIÓN

El presente trabajo, tiene como finalidad desarrollar el diseño y la implementación de un sistema de telemetría a una estación meteorológica impresa en 3D a través de un computador de placa reducida denominada *Raspberry Pi 3*, en el Instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología, INSIVUMEH. Haciendo uso de un módulo GPRS, el cual soporta la transmisión de datos, generados por sensores capaces de transformar las señales análogas, mediante fenómenos meteorológicos, en señales digitales para ser interpretadas, documentadas y visualizadas por la *Raspberry Pi 3*, las cuales son transferidas a un servidor de INSIVUMEH.

En el primer capítulo se presentan las generalidades del INSIVUMEH, identificando el área en donde se desarrolla el sistema. Seguidamente, en el capítulo dos se describen los fundamentos meteorológicos y microcontroladores que servirán de estudio, análisis y desarrollo de la estación meteorológica impreso en 3D. En el tercer capítulo se definen los diferentes sensores a utilizar y se describen sus características técnicas, funcionamiento y compatibilidad con la *Raspberry Pi 3*. En el capítulo cuarto se describe el diseño de cada sensor, módulo GPRS y base de datos, para el sistema de telemetría de la estación meteorológica.

Finalmente, en el capítulo cinco se desarrolla la implementación de cada uno de los componentes de la estación meteorológica, sus diagramas de flujo interconexiones físicas y el *software* necesario para la adquisición, procesamiento, base de datos, transmisión y visualización de los datos meteorológicos obtenidos.

El sistema propuesto surge de la necesidad de crear una estación meteorológica con tecnología vanguardista de bajo costo, que cumpla con las características técnicas de diseño y robustez necesarias para ser confiable ante la WMO (World Meteorological Organization por sus siglas en inglés).

1. GENERALIDADES DEL INSTITUTO NACIONAL DE SISMOLOGÍA, VULCANOLOGÍA, METEOROLOGÍA E HIDROLOGÍA

1.1. Historia

El instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología, también conocido por sus siglas como INSIVUMEH, es una institución técnico-científica que ejecuta actividades asociadas a las ciencias atmosféricas, geofísicas e hidrológicas.

Fue formado por el Observatorio Nacional dependiente del Ministerio de Agricultura con el departamento de Meteorología de la Dirección General de Aeronáutica Civil, dependiente del Ministerio de Comunicaciones y Obras Públicas (hoy en día Ministerio de Comunicaciones, Infraestructura y Vivienda) y la sección de Construcción y Mantenimiento de la red Hidrometeorológica del INDE.

Actualmente la institución técnico-científica es altamente calificada, la cual contribuye a la optimización de actividades del sector productivo de la República de Guatemala, asociada a las ciencias atmosféricas, geofísicas e hidrológicas; coordinando servicios con el sector privado y actuando como asesor técnico del gobierno en caso de desastres naturales.

Además, planifica, diseña y ejecuta estudio y monitoreo sistematizado con la tecnología adecuada, enriqueciendo las bases de datos y sistemas de

información geofísica referente del país, contribuyendo con la modernización y especificaciones del sector educativo.

Presta servicios de vigilancia sistemática permanente de la actividad volcánica y sísmica, manteniendo datos hidrometeorológicos, manteniendo, reactivando y realizando la ampliación de las actividades operacionales e institucionales en cuanto a la rehabilitación y reconstrucción de estaciones o puntos de monitoreo de carácter climático.

Su sede principal se encuentra en la ciudad de Guatemala, específicamente en la 7 av. 14-57 zona 13, colonia Aurora I; frente al aeropuerto internacional La Aurora, localizado geográficamente en las coordenadas 14 ° 35' 11" latitud, 90 ° 31' 58" longitud y elevación 1502 M.S.N.M.

Sus instalaciones son propias, modernas y funcionales, en donde se desarrollan convenientemente sus actividades científicas, de divulgación, capacitación y educativas.

Mantiene excelentes relaciones con instituciones de investigación y universidades nacionales y extranjeras. La institución es respetada y reconocida en el ámbito nacional e internacional como un instituto de vanguardia en tecnología aplicada en el campo de las ciencias de la tierra.

1.2. Misión

Ser una institución técnico-científica que genera y difunde información geocientífica, a través de la recolección y el procesamiento de datos para la toma de decisiones que contribuyan al beneficio de la población.

1.3. Visión

Ser reconocidos como una institución generadora de información geocientífica confiable, efectiva y de calidad mundial que impacte positivamente a la población.

1.4. Estructura de la institución

El INSIVUMEH es dependiente del Ministerio de Comunicaciones, Infraestructura y Vivienda; pero su eje principal, está compuesto por la dirección y subdirección general, de la cual se desglosan cuatro unidades complementarias y cuatro departamentos esenciales.

Las unidades complementarias de INSIVUMEH son las siguientes:

- Unidad de Planificación y Desarrollo Institucional.
- Unidad de Asesoría Jurídica.
- Unidad de Comunicación Social.
- Unidad de Tecnologías de la Información.

Cada unidad desarrolla actividades complementarias, las cuales son relacionadas a los distintos departamentos de la institución.

1.4.1. *Departamento Administrativo - Financiero*

Su función principal es la prestación de servicios técnico – administrativo y financiero indispensable para desarrollar las actividades de la institución.

Se subdivide en tres áreas, y cada área se desglosa en distintas unidades:

- Sección administrativa.
 - Unidad de compras.
 - Unidad de almacén.
 - Unidad de logística y servicios generales.
 - Unidad de mantenimiento de edificios.

- Sección financiera.
 - Unidad de contabilidad.
 - Unidad de presupuesto.
 - Unidad de tesorería.
 - Unidad de inventario.

- Sección de recursos humanos.
 - Unidad de nómina.
 - Unidad de acción de recursos humanos.
 - Unidad de información pública.

1.4.2. *Departamento de Investigación y Servicios Hídricos*

Desarrolla actividades de monitoreo y control en materia hídrica del territorio nacional, buscando como objetivo mantener actualizada la base de datos hidrometeorológica para el interés nacional; con la finalidad de optimizar el uso de fuentes hídricas disponibles en el territorio nacional, fomentando su conservación.

Se subdivide en distintas secciones, las cuales son:

- Sección de Hidrología y Aplicaciones.
- Sección de Hidrogeología.

- Sección de Oceanografía.
- Sección de Calidad del Agua y Aire.

1.4.3. Departamento de Investigación y Servicios Meteorológicos

Es el departamento encargado del monitoreo de fenómenos atmosféricos y su impacto a nivel nacional, generando reportes, estadísticas y pronósticos meteorológicos; difundiendo toda actividad meteorológica que se desarrolle en el país.

Se subdivide en distintas secciones, las cuales son:

- Sección de Climatología y Aplicaciones.
- Sección de Meteorología Aeronáutica.
- Sección Centro Nacional de Pronósticos.

1.4.4. Departamento de Investigación y Servicios Geofísicos

Departamento destinado al estudio de los movimientos telúricos. El cual tiene como actividad principal registrar los movimientos del suelo, su origen y propagación de la onda sísmica. Es comprendida por la unidad de vulcanología, la cual tiene como objetivo principal, registrar y monitorear la actividad volcánica en el territorio nacional.

Se subdivide en distintas secciones, las cuales son:

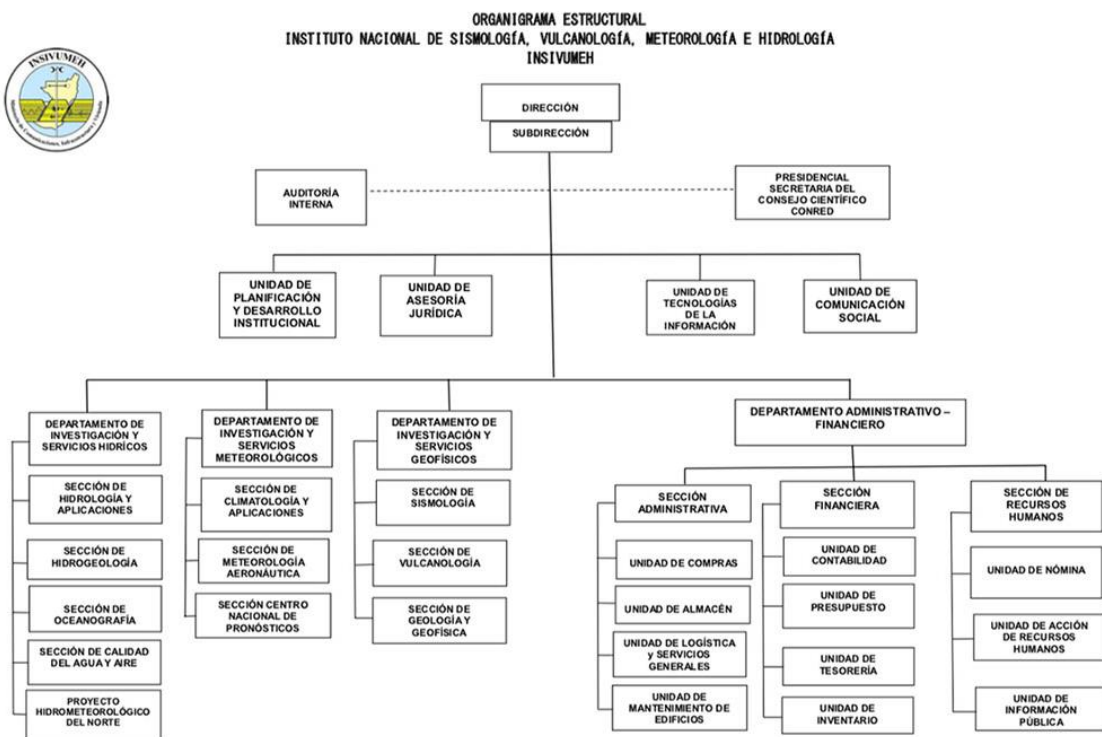
- Sección de Sismología.
- Sección de Vulcanología.

- Sección de Geología y Geofísica.

1.5. Organigrama

En la figura 1 se detalla la organización; por departamentos, secciones y unidades que conforma el Instituto Nacional de Sismología, Vulcanología, Meteorología e Hidrología.

Figura 1. Organigrama de INSIVUMEH



Fuente: INSIVUMEH. *Organigrama*. <https://insivumeh.gob.gt/institucional/quienes-somos/> Consulta: 09 de octubre de 2021.

2. TELEMETRÍA POR GPRS DE UNA ESTACIÓN METEOROLÓGICA

2.1. Estación meteorológica

La utilidad principal de una estación meteorológica es realizar mediciones y observaciones puntuales de los diferentes parámetros meteorológicos, utilizando los instrumentos adecuados para así poder establecer el comportamiento atmosférico.

Los datos recolectados por las estaciones meteorológicas permiten tener las siguientes funcionalidades:

- Conocer las condiciones meteorológicas de un sitio específico.
- Comparar información con otras estaciones aledañas a la estación de interés.
- Aportar información para realizar pronósticos meteorológicos basados en modelos numéricos.
- Crear información climática representativa del lugar donde se toman los datos.
- Crear alertas específicas ante fenómenos meteorológicos que pudieran ser de interés.
- Correlación de fenómenos meteorológicos con situaciones de riesgo, accidentes, destrucción de infraestructuras, entre otros.
- Información para la agricultura, la cual es uno de los principales sectores económicos del país.

2.1.1. Variables meteorológicas

Una variable meteorológica es una magnitud que permite adquirir distintos valores numéricos dentro de un conjunto de números especificado; las cuales son procesadas para obtener probabilidades de su comportamiento en un corto, mediano y largo plazo.

2.1.1.1. Temperatura

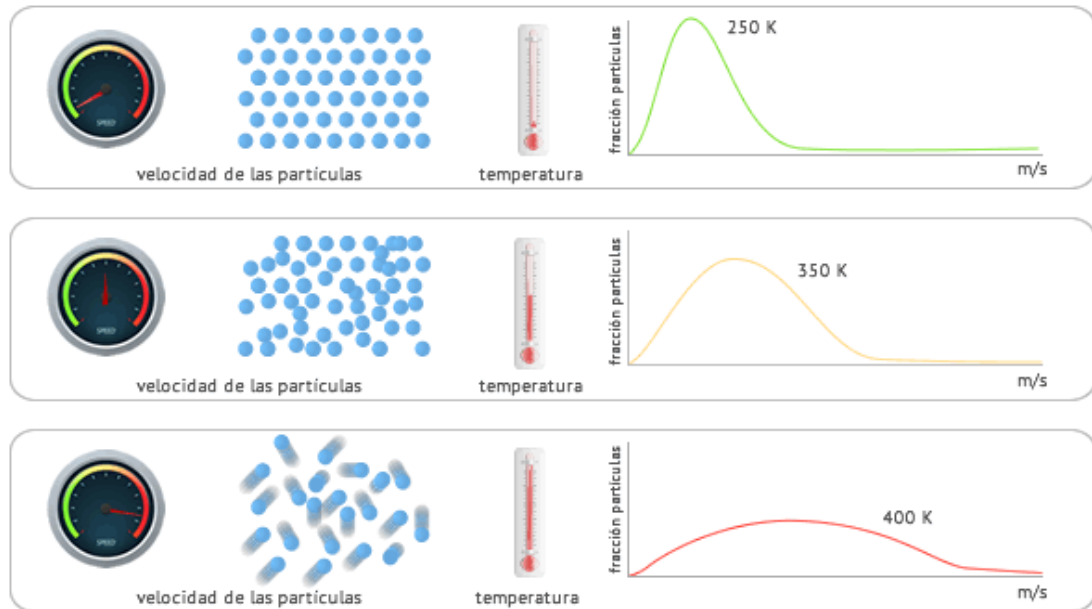
Es una magnitud escalar que se define como la cantidad de energía cinética de las partículas de una masa gaseosa, líquida o sólida.

Cuanto mayor es la velocidad de las partículas, mayor es la temperatura y viceversa.

La distribución de velocidades de las partículas de un gas (ya que en meteorología se registra la temperatura del aire) se rige por la ley de distribución de Maxwell, la cual se basa en el número de moléculas por unidad de velocidad.

Esto se puede ejemplificar en la siguiente figura.

Figura 2. Ley de distribución de Maxwell



Fuente: Física Lab. *Temperatura*. <https://www.fisicalab.com/apartado/temperatura>. Consulta: 10 de octubre de 2021.

En la figura 2 se ejemplifica que, para una misma sustancia, cuanto mayor es la temperatura, mayor es la velocidad de las partículas que la componen.

En meteorología la medición de la temperatura debe excluir la radiación calorífica de los objetos que se encuentran en el ambiente (principalmente alrededor del equipo de medición), así como la eliminación de la velocidad del viento y la humedad relativa; a esto se le denomina temperatura seca.

Para optimizar el proceso de recolección de datos de temperatura seca, existen equipos de protección para el instrumento de medición, las cuales pueden ser casetas de madera, si es medición convencional o análoga, o plástico, como

el material ASA, que es usado generalmente en instrumentos digitales; a las que se les denomina pantallas de protección de Stevenson, las cuales son, principalmente, de color blanco, con el objetivo de que el material absorba la menor cantidad de radiación solar.

Las pantallas de *Stevenson* proveen protección contra la radiación solar, precipitación, viento, rocío y hielo.

Figura 3. **Pantalla Stevenson**



Fuente: elaboración propia. INSIVUMEH.

Existen distintos tipos de escalas para medir la temperatura, las más comunes son:

- Escala Celsius ($^{\circ}\text{C}$): conocida también como “escala centígrada”, debido al intervalo de cien grados entre los puntos definidos. Esta escala

pertenece al Sistema Internacional de Unidades, creada en 1742 por Anders Celsius.

- Escala Fahrenheit (°F): fue propuesta por Gabriel D. Fahrenheit, y data del año 1714; y es utilizada en la mayoría de los países de habla inglesa. El punto de congelación del agua ocurre a los 32 °F y su punto de ebullición a los 212 °F.
- Escala Kelvin (K): propuesta por Lord Kelvin, en el año 1848. Dicha escala no cuenta con números negativos de temperatura, debido a que su cero está situado en el estado donde las partículas que forman el material no poseen ningún movimiento. Es la escala más utilizada por los científicos a nivel mundial.

2.1.1.2. Presión atmosférica

Es la fuerza que ejerce el peso del aire sobre la superficie terrestre, y sobre todos los seres que en ella habitan; está definida como una fuerza por unidad de superficie. A lo cual se le asocia una unidad de medida, que en el Sistema Internacional es el Pascal (1 Pascal = 1 N/m²).

La presión atmosférica en un punto coincide numéricamente con el peso de una columna estática de aire de sección recta unitaria, que se extiende desde ese punto hacia el límite superior de la atmósfera. Como la densidad del aire disminuye conforme aumenta la altura, no se puede calcular ese peso, a menos que seamos capaces de expresar la variación de la densidad del aire ρ en función de la altitud z o de la presión p .

La presión atmosférica en un lugar determinado experimenta variaciones asociadas con los cambios meteorológicos; por otra parte, en un lugar determinado, la presión atmosférica disminuye con la altitud.

El instrumento para realizar mediciones de presión atmosférica se denomina barómetro. En la actualidad existen una gran variedad de este dispositivo en el mercado, los cuales pueden ser de tipo capacitivo, resistivo, entre otros.

Por los sistemas de medición, existen distintas unidades de medida para la presión atmosférica, siendo las más comunes: milibares, atmosferas, pascales, mm de mercurio.

Las conversiones entre unidades de medida de la presión atmosférica son las siguientes:

$$1 \text{ atm} = 760 \text{ mmHg} = 101\,300 \text{ N/m}^2 \text{ o (Pa)} = 1\,013 \text{ mb.}$$

Figura 4. **Barómetro**



Fuente: Nautica Merino. *Barómetro náutico*. https://nauticamerino.com/267-medium_default/barometro-nautico-b72n.jpg. Consulta: 10 de octubre de 2021.

2.1.1.3. *Humedad*

La humedad se define como la cantidad de vapor de agua que está contenida en el aire. Dicha cantidad no es constante, ya que depende de varios factores, por ejemplo, la cantidad de precipitación acumulada en el suelo, la cercanía al mar, las plantas, temperatura, entre otros fenómenos.

Para hacer referencia al contenido de humedad en la atmósfera, existen distintos conceptos:

- Humedad absoluta: masa de vapor de agua en gramos, que contiene un metro cúbico de aire seco.
- Humedad específica: masa de vapor de agua en gramos, que contiene un kilogramo de aire.
- Razón de mezcla: masa de vapor de agua en gramos, que contiene un kilogramo de aire seco.
- Humedad relativa: contiene una masa de aire, en relación con la máxima humedad absoluta que podría admitir sin producirse condensación, conservando las mismas condiciones de temperatura y presión atmosférica.

En meteorología se usa de manera usual la medida de humedad relativa, la cual se expresa en tanto por ciento (%), y se calcula según la siguiente expresión.

$$h = \frac{e}{E} * 100$$

En donde e define el contenido de vapor de la masa de aire y E la máxima capacidad de almacenamiento de dicha masa, denominada presión de vapor

saturante. Este valor indica la cantidad de vapor de agua que una masa de aire puede contener antes de transformarse en agua líquida. De alguna manera la humedad relativa expresa una idea de lo cerca que esta una masa de aire para alcanzar el punto de saturación.

Un valor de 100 % de humedad relativa, indica que la masa de aire no puede almacenar más vapor de agua y, por tanto, un mínimo más de este se convertirá en agua o en cristalinicos de hielo, dependiendo de las condiciones ambientales.

Entre los diversos equipos de medición de la humedad, existe el higrómetro que realiza la medición directa y permite observar de manera directa la humedad relativa; el psicrómetro mide la diferencia entre un termómetro seco y un termómetro húmedo.

Figura 5. **Psicrómetro digital**



Fuente: Testers. *Psicrómetro digital*. https://www.testers.com.ve/wp-content/uploads/2021/03/2998_89221_p1-768x768.jpg. Consulta: 10 de octubre de 2021.

2.1.1.4. Precipitación

La precipitación se define como cualquier forma de hidrometeoro que cae de la atmósfera y llega a la superficie terrestre. Para que este proceso pueda llevarse a cabo es necesario que el vapor existente en una masa de aire, que ha alcanzado el grado de saturación, puede condensarse en forma de gota, debe cumplir con dos condiciones básicas:

- La masa de aire debe alcanzar una temperatura de enfriamiento suficiente para producir la precipitación.
- En el aire deben existir núcleos de condensación, denominados núcleos higroscópicos, sobre los cuales se pueden formar las gotas de agua.

Una vez formada la nube, las diminutas gotas que la forma deben estar suspendidas dentro de la misma, a causa de las corrientes existentes de forma ascendente, estas empezarán a aumentar de tamaño a causa de otras gotas que se encuentran durante la caída. Sobre cada gota en la nube actúan dos fuerzas fundamentales, la que ocasiona el arrastre de la corriente de aire ascendente y la generada por el peso de la gota.

Cuando el peso de la gota formada en el interior de la nube supera la fuerza de arrastre caerá hacia el suelo, produciendo la lluvia. Las gotas alcanzan un tamaño mayor cuanto más tiempo estén dentro de la nube ascendiendo y descendiendo, y cuanto más ganen contenido de agua líquida que se pueda suministrar la nube.

Dependiendo del tamaño que lleguen a tener las gotas que caen a la superficie terrestre y de la manera que estas caigan, se puede tener diferentes tipos de precipitación:

- Precipitación líquida: llovizna (gotas pequeñas que caen de manera uniforme), lluvia (gotas de mayor tamaño que caen de manera violenta e intensa).
- Precipitación glacial: llovizna congelada, lluvia congelada.
- Precipitación congelada: nieve, granos de nieve, bolas pequeñas de hielo, granizo, copos de nieve y cristales de hielo.

Figura 6. **Proceso de crecimiento de una gota en el interior de una nube**



Fuente: Universidad de Salamanca. *Una gota de nube.*

https://rsef.es/images/Fisica/P1_Lagota.pdf. Consulta: 10 de octubre de 2021.

El equipo de medición utilizado para la precipitación en un área determinada dentro de un intervalo de tiempo se denomina pluviómetro. Dicho dispositivo está integrado por un vaso recolector, en forma de embudo profundo, que envía el agua a un recipiente graduado con forma de balancín, donde se acumula el total de agua caída, una vez llega a su máximo nivel, este es volcado, vaciando el agua para iniciar el ciclo de medición nuevamente y genera una señal, el cual se representa una medición de litros por metro cuadrado o el equivalente en

milímetros (mm). Esta medida indica la altura en milímetros que alcanzaría una capa de agua que cubriese una superficie horizontal equivalente a un metro cuadrado. A este instrumento se le denomina pluviómetro.

Figura 7. **Pluviómetro**



Fuente: SENSOVANT. *Pluviómetro*.

<https://th.bing.com/th/id/OIP.JfMcdEn4ejg8UZdkKbiXAwAAAA?pid=ImgDet&rs=1>. Consulta: 10 de octubre de 2021.

2.1.1.5. Viento

El viento se denomina al flujo de gases a gran escala que es resultado de grandes corrientes de convección en la atmósfera de la Tierra, impulsadas por la energía térmica solar. Las diferencias en la presión atmosférica causan vientos de variadas velocidades, pues cuando existe algún cambio en tal presión, el aire se mueve desde la zona de presión más alta hacia la más baja.

El viento está presente de forma natural y posee un gran potencial para el aprovechamiento humano.

El viento puede clasificarse según su escala espacial, su velocidad y los tipos de fuerzas que lo ocasionan, las zonas en donde se produce y sus efectos.

En meteorología, el viento, se pueden ser ráfagas si son de alta velocidad y turbonadas si son fuertes y duran aproximadamente 1 minuto. De acuerdo con su fuerza media, los vientos de larga duración se consideran brisa, vendaval, tormenta, huracán o tifón.

La escala de *Beaufort* se encarga de dividir el viento en 17 niveles, con base en su velocidad. Por ejemplo, si la velocidad de un viento es de 0 a 1 km/h, se denomina calma. Si es de 62 a 74 km/h es un temporal, y si su velocidad está por encima de los 118 km/h, corresponde a un huracán.

Figura 8. **Escala de *Beaufort***

			<i>Km/hora</i>	<i>Nudos</i>	
3	Flojo (Brisa débil)		12 a 19	7 a 10	Se agitan las hojas, ondulan las banderas
4	Bonancible (Brisa moderada)		20 a 28	11 a 16	Se levanta polvo y papeles, se agitan las copas de los árboles
5	Fresquito (Brisa fresca)		29 a 38	17 a 21	Pequeños movimientos de los árboles, superficie de los lagos ondulada
6	Fresco (Brisa fuerte)		39 a 49	22 a 27	Se mueven las ramas de los árboles, dificultad para mantener abierto el paraguas
7	Frescachón (Viento fuerte)		50 a 61	28 a 33	Se mueven los árboles grandes, dificultad para andar contra el viento
8	Temporal (Viento duro)		62 a 74	34 a 40	Se quiebran las copas de los árboles, circulación de personas dificultosa
9	Temporal fuerte (Muy duro)		75 a 88	41 a 47	Daños en árboles, imposible andar contra el viento
10	Temporal duro (Temporal)		89 a 102	48 a 55	Árboles arrancados, daños en la estructura de las construcciones
11	Temporal muy duro (Borrasca)		103 a 117	56 a 63	Estragos abundantes en construcciones, tejados y árboles
12	Temporal huracanado (Huracán)		118 y más	64 a 71 >	Destrucciones abundantes, estragos y lluvias

Fuente: imgur. *Escala de Beaufort*. <https://i.imgur.com/tYHuQ9d.jpeg>. Consulta: 10 de octubre de 2021.

Para tener una referencia de medidas directas de la velocidad y dirección del viento, en meteorología se utilizan diferentes instrumentos de medición:

- Anemómetro: instrumento utilizado para medir la velocidad horizontal del viento, donde la cantidad de giros de las cazoletas es proporcional a la velocidad del viento. La unidad de medida utilizada comúnmente es km/h o m/s.

Figura 9. **Anemómetro**



Fuente: elaboración propia. INSIVUMEH.

- Veleta: es el instrumento utilizado para medir la dirección del viento, el cual proporciona la procedencia geográfica del viento. En este sentido se habla de viento norte, noreste, este, sureste, sur, suroeste, oeste y noroeste.

Figura 10. **Veleta**

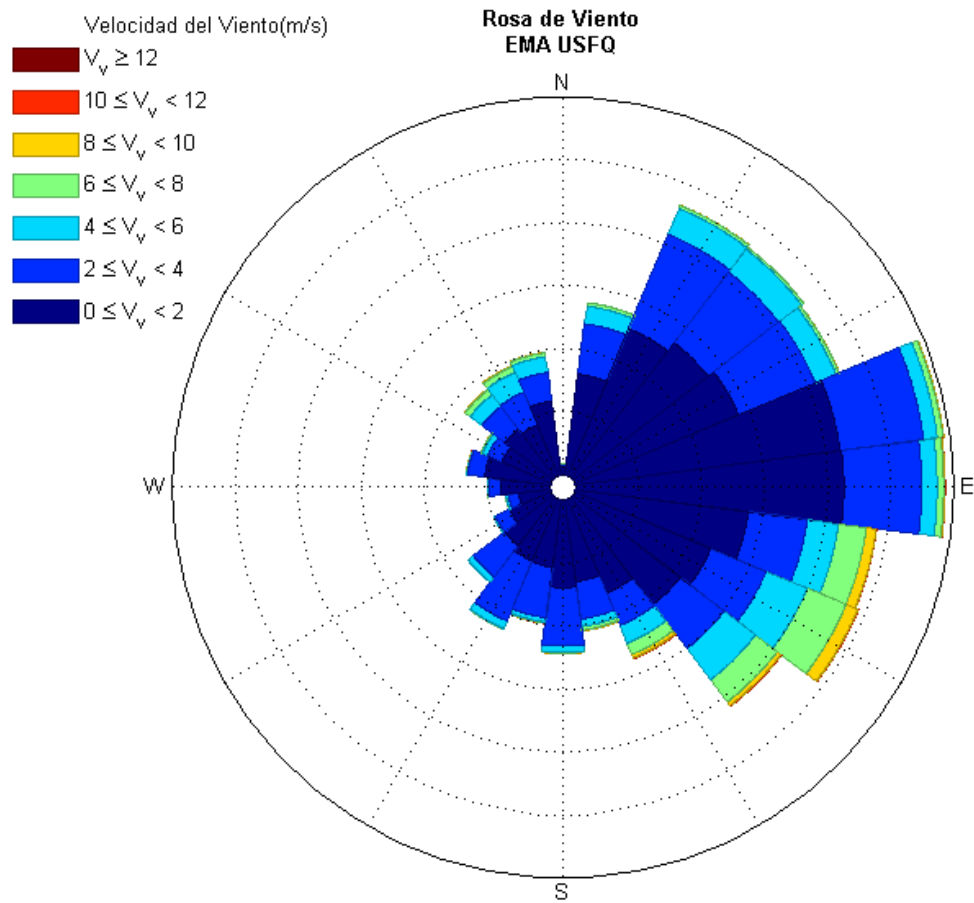


Fuente: elaboración propia. INSIVUMEH.

Para determinar la dirección del viento se utiliza comúnmente la rosa de vientos, la cual determina el nombre del viento en función de los puntos cardinales.

La rosa de vientos es un símbolo en forma de círculo y tiene marcado los puntos en que se divide el horizonte, data del siglo XV atribuyendo su invención a Ramon Llull.

Figura 11. **Rosa de vientos**



Fuente: CAZORLA, María. *Rosa de viento*. https://www.researchgate.net/figure/Figura-9-Rosa-de-viento-preparada-con-datos-de-un-minuto-del-7-al-20-de-Enero-de-2015-en_fig4_284070945. Consulta: 10 de octubre de 2021.

2.1.1.6. **Radiación solar**

La radiación solar es la energía emitida por el Sol, que se propaga en todas las direcciones a través del espacio mediante ondas electromagnéticas. Esta energía es el motor que determina la dinámica de los procesos atmosféricos y el clima.

La energía procedente del sol es radiación electromagnética proporcionada por las reacciones del hidrógeno en el núcleo del Sol por fusión nuclear y emitida por la superficie solar.

Medir este fenómeno es importante para un amplio rango de aplicaciones en las áreas de ingeniería, arquitectura, agricultura, ganadería, salud humana y meteorología; dentro de las cuales se destacan el análisis de la evaporación e irrigación, su importante rol en los modelos de calidad del aire y de predicción del tiempo y clima.

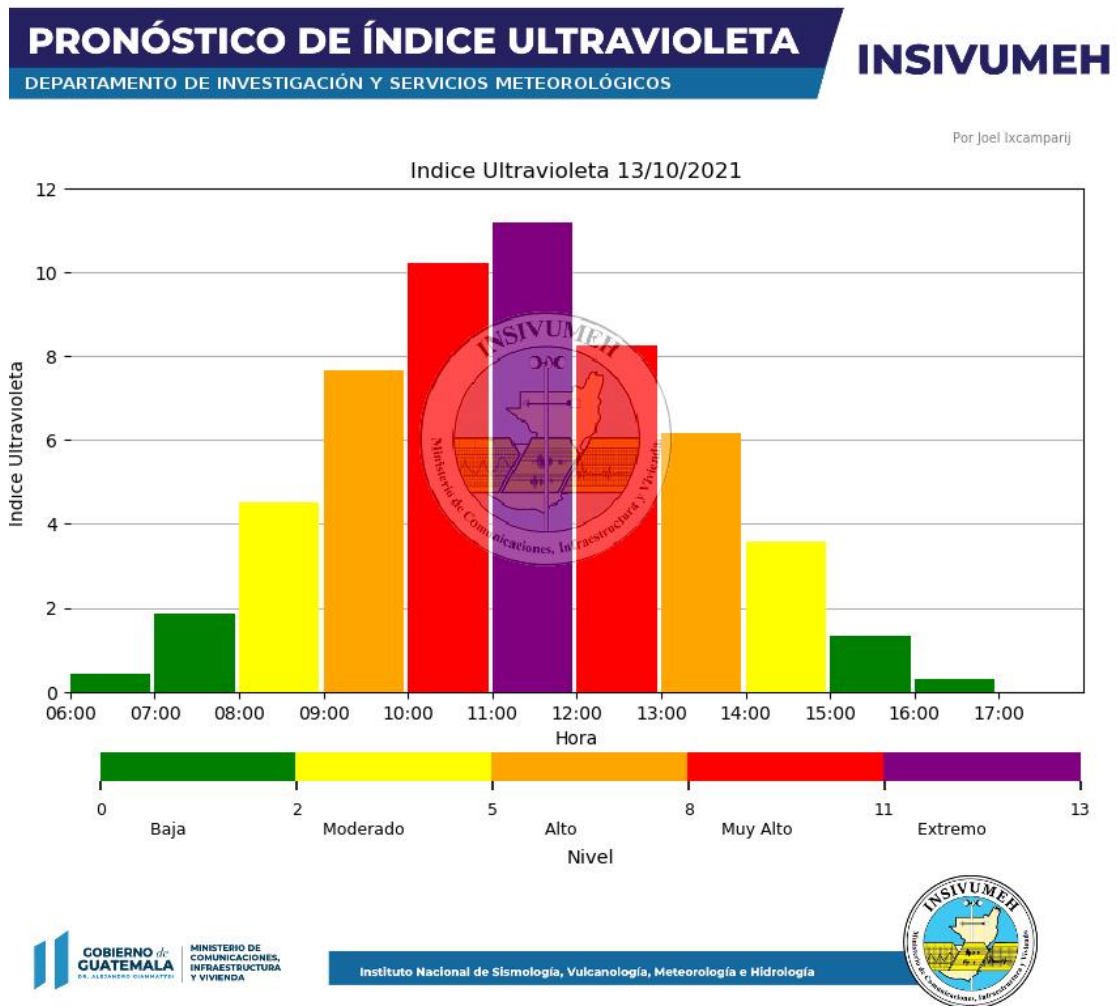
La radiación es emitida sobre un espectro de longitud de ondas, con una cantidad específica de energía para cada longitud de onda. La longitud de onda de la luz es tan corta que suele expresarse en nanómetros (nm), que equivalen a una milmillonésima de metro.

La radiación electromagnética se puede ordenar en un espectro en diferentes longitudes de onda, como se muestra en la figura 12, donde se extiende desde longitudes de onda corta de billonésimas de metro (frecuencias muy altas), hasta longitudes de onda larga de muchos kilómetros (frecuencias muy bajas). El espectro electromagnético no tiene definidos límites superior ni inferior, y la energía de una fracción diminuta de radiación, llamada fotón, es inversamente proporcional a su longitud de onda, es decir, a menor longitud de onda mayor contenido energético.

Para uso meteorológico, también se pueden utilizar pronósticos de índice de UV, el cual se puede complementar con los valores obtenidos por sensores de radiación solar. El índice ultravioleta (UV) es un método para predecir la intensidad con la que la radiación UV solar alcanza la superficie terrestre,

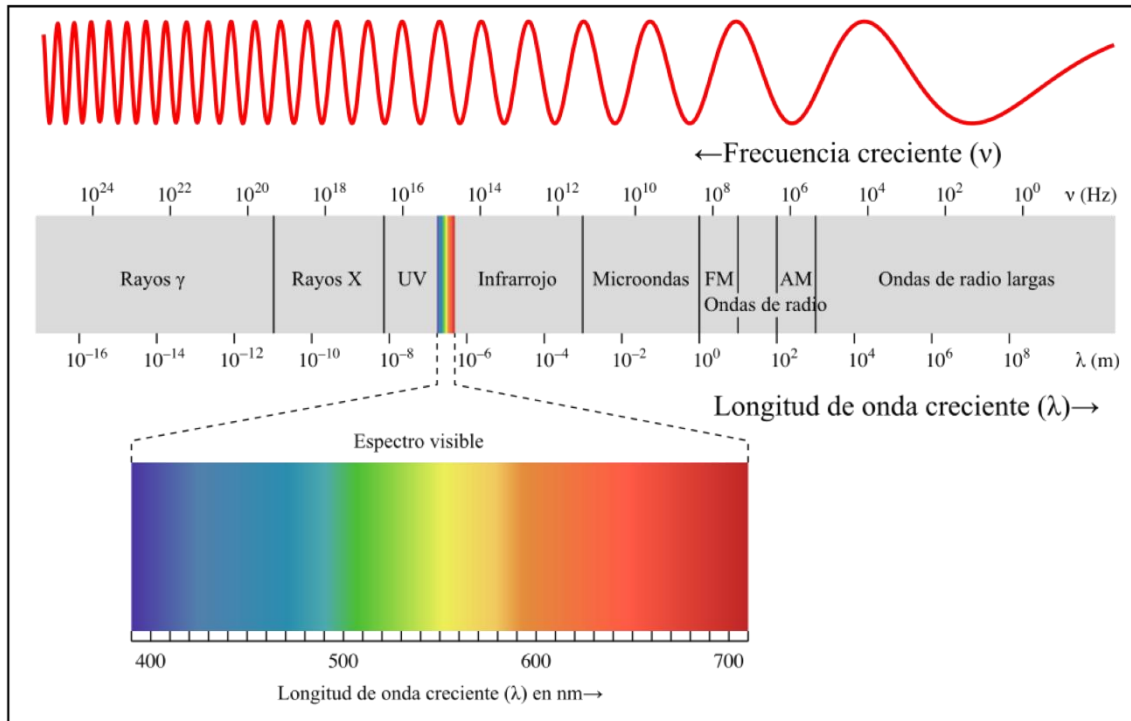
orientándonos asó sobre las precauciones que el sector de interés debe tomar para evitar una sobreexposición.

Figura 12. Pronóstico de índice ultravioleta



Fuente: IXCAMPARIJ, Joel. *Sensores Remotos y Monitoreo Atmosférico*.
http://www.insivumeh.gob.gt/img/FUEGO/fuego_preview/main.pdf. Consulta: 13 de octubre de 2021.

Figura 13. Espectro electromagnético



Fuente: Quora. *Espectro electromagnético*.

https://th.bing.com/th/id/R.2284774f79143aa65f5e52ef998b0e0c?rik=4%2bxFzR3TxzpHdg&riu=http%3a%2f%2fwww.gaiaciencia.com%2fwp-content%2fuploads%2f2016%2f11%2fspectrum_EM_1.png&ehk=KSGsyvcAuBYGsVJa%2bS8nCUli1LLhIBctUcwmkp2a7J0%3d&risl=&pid=ImgRaw&r=0. Consulta: 10 de octubre de 2021.

En meteorología es de gran valor los datos correspondientes a la luz visible durante un día, a los que se le denomina horas solares, para ello es necesario determinar la hora exacta en la que el espectro de luz visible es apreciado durante la mañana, y la puesta del sol durante la tarde, dando paso a la noche.

Este dato se obtiene con la hora local y es variable durante el año, debido al movimiento de traslación de la tierra.

El instrumento de medición meteorológico para la radiación solar es conocido como piranómetro, el cual cuantifica inequívocamente la radiación solar orientado a la ocurrencia en la superficie de la Tierra. Es un sensor destinado a cuantificar el espesor de la corriente de radiación solar en un campo de 180 grados. El piranómetro es un sensor que mide la radiación global de onda corta.

Figura 14. **Piranómetro**



Fuente: TH Bing. *Piranómetro*.

[Thttps://th.bing.com/th/id/R.20744103ce0abece8423ab362559b850?rik=CBYL%2bRhEf46NXg&pid=ImgRaw&r=0](https://th.bing.com/th/id/R.20744103ce0abece8423ab362559b850?rik=CBYL%2bRhEf46NXg&pid=ImgRaw&r=0). Consulta: 10 de octubre de 2021.

2.1.1.7. Telemetría

La telemetría facilita la monitorización a distancia de procesos o equipos, el envío de alertas o alarmas a un centro de control o plataforma de visualización en internet, con el fin de que el funcionamiento sea seguro y eficiente.

La telemetría es una tecnología que permite la medición y control de variables físicas o químicas a través de datos que son transferidos a una central de control.

El envío de información hacia el operador en un sistema de telemetría se realiza normalmente mediante comunicación inalámbrica, aunque también puede realizar por otros medios (teléfono, redes de computadoras, enlaces de fibra óptica, entre otros). Los sistemas de telemetría reciben las instrucciones y los datos necesarios para operar desde un centro de control o desde una plataforma de visualización en internet.

2.1.2. Funcionamiento

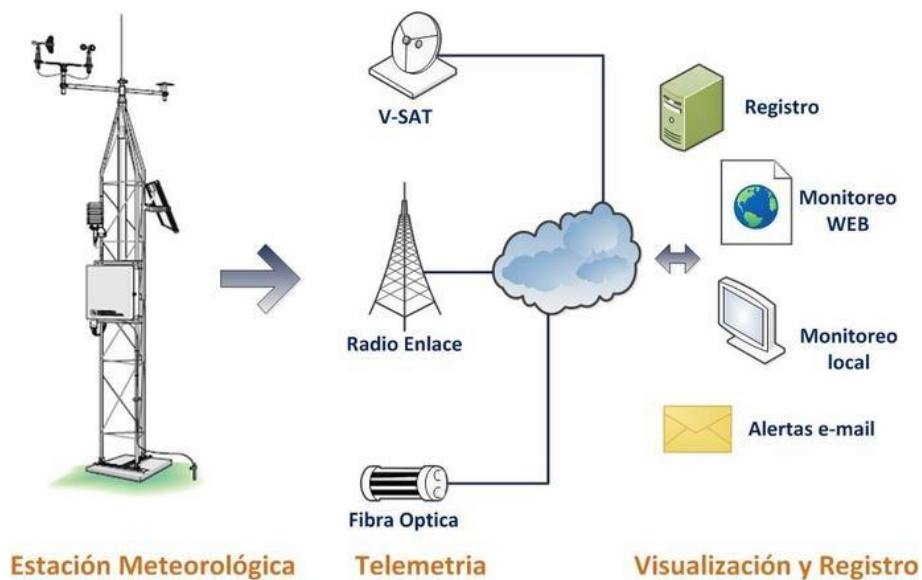
La telemetría se basa en cuatro etapas:

- **Medición:** realiza la captura y procesamiento de información en la fuente de generación (proceso o máquina) a través de sensores y/o hardware.
- **Comunicación:** es la transmisión de datos adquiridos del proceso a través de radiofrecuencia, celular, satélite, LoRa, entre otros.
- **Visualización:** los datos son almacenados en servidores virtualizados (nube) y presentados a través de plataformas de visualización.
- **Analítica:** es el uso de plataformas de inteligencia de negocios que facilitan la interpretación de los datos y la toma de decisiones.

La telemetría permite supervisar fenómenos atmosféricos como la presión, temperatura, viento, gases, entre otros; del mismo modo, prevé cuando puede ocurrir un desastre natural, a través de telemetría por radio, que mide el comportamiento de ondas y tamaños.

Entre los beneficios y ventajas principales de la telemetría, los sensores en procesos permiten recopilar y generar constantemente datos que, analizados óptimamente, permiten automatizar procesos en la gestión de toma de decisiones basados en situaciones y comportamientos reales.

Figura 15. **Telemetría**



Fuente: Telemetic. *Telemetría*.

https://www.telemetic.com.mx/uploads/3/1/3/4/31348455/413726625_1.jpg. Consulta: 10 de octubre de 2021.

2.2. GPRS

Servicio general de paquetes vía radio, conocido como GPRS por sus siglas en inglés, fue lanzado en 2001 por la red GSM para proporcionar acceso a internet a los usuarios de dispositivos móviles.

Por tratarse de una tecnología de datos inalámbricos, GPRS ofrece velocidades de datos máximas de 115 Kbps. GPRS es una tecnología basada en paquetes, lo que significa que los datos están divididos en paquetes que se transmiten en breves ráfagas sobre una red IP. Este diseño es mucho más eficiente que las redes conmutadas por circuitos, dando lugar a una reducción de los costos operativos de la red. El diseño de paquetes beneficia a los usuarios en dos formas primordiales. Primero, GPRS provee una conexión siempre activa, que no exige al usuario conectarse cada vez que desea obtener acceso a datos; y como segundo punto, los usuarios solo pagan por la cantidad de datos necesarios para su transmisión de paquetes.

Los operadores de servicio GPRS tienen la oportunidad de desarrollar y lanzar avanzados servicios de datos mucho más rápidamente y de manera menos costosa, beneficio que se atribuyen a equipos ofrecidos en versiones estándar. GPRS es la tecnología inalámbrica de datos en paquetes más ampliamente soportada en el mundo.

Al igual que GSM, GPRS soporta *Roaming* imperceptible al usuario, permitiendo que los usuarios tengan acceso a sus servicios de datos mientras se encuentran de viaje.

Existen variedad de módulos GSM/GPRS, pero en este proyecto nos enfocaremos en el SIM900.

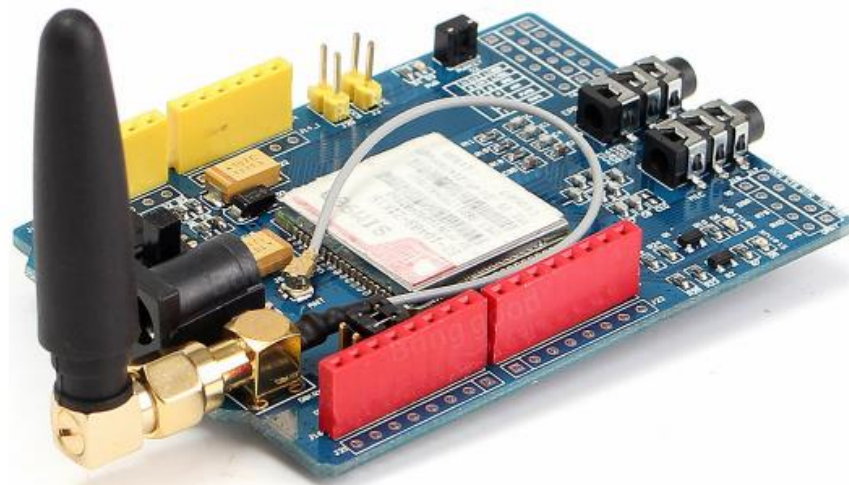
2.2.1. Módulo GPRS/GSM SIM900

El módulo SIM900 es una tarjeta ultra compacta y confiable; compatible con distintos microcontroladores. Permite trabajar en frecuencias desde 850 MHz hasta 1 900 MHz, para realizar llamadas de voz, envíos de sms y fax.

Permite ser controlado y configurado mediante protocolo UART, usando comandos AT. Por default la velocidad UART está establecida en 19200 baudios. Sus especificaciones son:

- Voltaje de alimentación mínima 9V.
- Voltaje de alimentación máxima 12V.
- Corriente 1,5 mA.
- Comunicación UART.
- Bandas de frecuencia 850/900/1 800/1 900 MHz.
- Servicio de mensajería corta.
- Pila embebida TCP/UDP.
- Temperatura de operación mínima -40 °C.
- Temperatura de operación máxima 85 °C.

Figura 16. **Módulo GPRS/GSM SIM900**



Fuente: CDN Shopify. *Módulo GPRS/GSM SIM900.*

https://cdn.shopify.com/s/files/1/0409/9041/products/sim_ab621448-9c61-4308-a89e-a7630875aeb8_1024x1024.png?v=1615494936. Consulta: 10 de octubre de 2021.

2.2.2. Comandos AT

Los comandos AT son utilizados para la gestión de módems. Consiste en una cadena de caracteres que comienzan con las letras at, y culminan en los signos de retorno; sin embargo, en la cadena de inicialización que se cierran los elementos se generan automáticamente. Los símbolos de los comandos AT pueden ser en mayúsculas o minúsculas. Entre los distintos equipos pueden plantearse los espacios en blanco. Ejecutan las principales acciones relacionadas con la conmutación de modos de módem, los comandos AT pueden controlar la comunicación del dispositivo. Ofrece diversa información que sea necesaria para garantizar el estado del módem; en donde puede almacenar las constantes que se definen en tiempo de espera, tipos de protocolos utilizados, señales analógicas, niveles de entrada y salida.

Tabla I. Comandos AT principales

Commands ↕	Description ↕	Type ↕	Set/Execute ↕	Inquiry ↕	test ↕	Parameters ↕
AT+RST	restart the module	basic	-	-	-	-
AT+CWMODE	wifi mode	wifi	AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	1= Sta, 2= AP, 3=both
AT+CWJAP	join the AP	wifi	AT+ CWJAP = <ssid>,<pwd >	AT+ CWJAP?	-	ssid = ssid, pwd = wifi password
AT+CWLAP	list the AP	wifi	AT+CWLAP			
AT+CWQAP	quit the AP	wifi	AT+CWQAP	-	AT+CWQAP=?	
AT+ CWSAP	set the parameters of AP	wifi	AT+ CWSAP= <ssid>,<pwd>,<chl>,<ecn>	AT+ CWSAP?		ssid, pwd, chl = channel, ecn = encryption
AT+ CIPSTATUS	get the connection status	TCP/IP	AT+ CIPSTATUS			
AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART= <type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART= <id> <type>,<addr>,<port>	-	AT+CIPSTART=?	id = 0-4, type = TCP/UDP, addr = IP address, port= port
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND= <length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND= <id>,<length>		AT+CIPSEND=?	
AT+CIPCLOSE	close TCP or UDP connection	TCP/IP	AT+CIPCLOSE=<id> or AT+CIPCLOSE		AT+CIPCLOSE=?	
AT+CIFSR	Get IP address	TCP/IP	AT+CIFSR		AT+ CIFSR=?	
AT+ CIPMUX	set mutiple connection	TCP/IP	AT+ CIPMUX=<mode>	AT+ CIPMUX?		0 for single connection 1 for multiple connection
AT+ CIPSERVER	set as server	TCP/IP	AT+ CIPSERVER= <mode>[,<port>]			mode 0 to close server mode, mode 1 to open; port = port

Fuente: i2. WP. *Comandos AT*. <https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2015/01/commands.png>. Consulta: 10 de octubre de 2021.

En la figura 16 se muestran los principales comandos AT que son enviados por UART al SIM900 para establecer comunicaciones de transmisión y recepción de datos para el protocolo TCP/IP.

2.2.3. APN

El APN (*Access Point Name*) es el punto de acceso a Internet de los dispositivos móviles; con el él le indicamos al teléfono el camino que se debe

seguir para conectarse a Internet, aunque también permite el envío y recepción de mensajes multimedia o MMS.

Con la configuración de APN un dispositivo crea una conexión a la puerta de enlace de un operador. Parte de este proceso implica que el operador use una red APN definida para elegir las direcciones IP asignadas y la configuración de seguridad cuando corresponda.

A través del APN se definen distintos parámetros:

- El nombre del punto de acceso.
- El nombre de usuario que hace la petición de conexión.
- La clave de los operadores móviles para configurar el acceso a Internet.

2.2.3.1. Tipos de APN

Hay dos tipos generales de APN, aquellos que se conectan a la red pública y los que se conectan a la red privada. También las direcciones IP estáticas y dinámicas se establecen en las APN públicas y privadas. Las direcciones IP dinámicas pueden cambiar con cada sesión de datos, mientras que las direcciones IP estáticas son reclamadas por un dispositivo.

2.2.4. Microcontroladores

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran con controlador y que contiene todos los componentes fundamentales de un ordenador, aunque de limitadas prestaciones y que se suele destinar a gobernar una sola tarea. En su memoria sólo reside un programa que controla en funcionamiento de una tarea determinada, sus líneas de entrada/salida se conectan a los sensores y

actuadores del dispositivo a controlar y, debido a su pequeño tamaño, suele ir integrado en el propio dispositivo al que gobierna.

Son diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación.

Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidores de señales analógicas a digitales, temporizadores, UART y buses de interfaz serie especializados como I2C. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados. Los modernos microcontroladores frecuentemente incluyen un lenguaje de programación. Estos dispositivos negocian la velocidad y la flexibilidad para facilitar su uso.

2.2.4.1. Historia

El microcontrolador fue inventado por *Texas Instruments* en la década de 1970, casi al mismo tiempo que el primer microprocesador que estaba siendo inventado en Intel. Los primeros microcontroladores eran simplemente microprocesadores con una función de memoria, como la memoria RAM y ROM. Más tarde, los microcontroladores se desarrollaron en una amplia gama de dispositivos diseñados para aplicaciones de sistemas integrados específicos en dispositivos tales como automóviles, teléfonos móviles y electrodomésticos.

En 1971, el primer microcontrolador fue inventado por dos ingenieros, Gary Boone y Michael Cochran, crearon el TMS 1000, el cual era un microcontrolador de 4 bits con función de ROM y RAM. El microcontrolador era utilizado internamente en *Texas Instruments* en sus productos de cálculos desde 1972

hasta 1974, y fue refinado con el paso de los años. El TMS 1 000 estuvo disponible en varios tamaños de RAM y ROM.

Figura 17. **TMS 1 000**



Fuente: TH. Bing. *TMS 1000*.

<https://th.bing.com/th/id/R.bd108484a51076e89d734820b3ebcb48?rik=IXZ2dO%2b5uv2H2A&pid=ImgRaw&r=0>. Consulta: 10 de octubre de 2021.

2.2.4.2. *Raspberry Pi*

La *Raspberry Pi* es un computador de bajo costo y un tamaño compacto, altamente funcional capaz de ejecutar un sistema operativo y diversos sistemas de programación; también puede ser utilizada como servidor. Es capaz de interactuar directamente con sensores, procesar datos y mostrarlos mediante una interfaz, es por ello que se desarrolla el presente trabajo por medio de la tecnología del microcontrolador *Raspberry Pi* ofrece.

El primer prototipo fabricado de *Raspberry Pi* es basado en una arquitectura ARM, fue diseñado en un paquete de tamaño comparable al de una memoria USB, y contaba en un extremo de un puerto USB; en el otro puerto un HDMI.

La popularidad que ha recibido *Raspberry Pi* ha sido principal por su bajo costo, versatilidad y facilidad de modificación para diferentes proyectos, y la capacidad de ejecutar un sistema operativo, siendo el más común Linux, el cual es un sistema muy popular entre los desarrolladores por ser de *software* libre.

2.2.4.3. *Raspberry Pi 3 B+*

Existen diferentes versiones y modelos de *Raspberry Pi*, pero en este trabajo se implementará *Raspberry Pi 3 B+*, el cual al inicio del desarrollo de este trabajo era la versión más reciente que se podía encontrar en el mercado guatemalteco.

2.2.4.4. *Software operativo de la Raspberry Pi*

El *Raspberry Pi* usa principalmente sistemas operativos GNU/LINUX. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de *Raspberry Pi*, se lanzó durante julio del año 2012 y es la distribución recomendada por la fundación.

A la GPU se accede mediante una imagen del *firmware*, que se carga dentro de la GPU al arrancar desde la tarjeta SD. El archivo está asociado a los controladores del núcleo Linux. Las aplicaciones hacen llamadas a las bibliotecas de tiempo de ejecución que son de código abierto, y las mismas hacen llamadas a unos controladores de código abierto en el núcleo Linux. La API del controlador del núcleo es específica para estas bibliotecas.

Para el desarrollo del proyecto en curso se utiliza el sistema operativo oficial, *Raspbian*, ya que cuenta con soporte oficial en su página Web.

La organización de *Raspberry Pi* desarrollo una herramienta para realizar la instalación del sistema operativo denominada NOOBS, que incluye los sistemas operativos oficiales, incluyendo *Raspbian*, el cual se utiliza en el desarrollo del presente proyecto, por su versatilidad, documentación en sitios oficiales y permite la utilización de PPP (protocolo punto a punto).

Para la implementación del proyecto, es necesario descargar la versión del sistema operativo desde el sitio oficial: <https://www.raspbian.org/RaspbianInstaller>, y realizar el proceso que se describe en la documentación oficial disponible en la misma página.

Es necesario disponer de una tarjeta microSD, con capacidad mínima de 32 GB, específicamente para este proyecto, ya que se estará realizando un almacenamiento interno, como respaldo de la información recolectada por los distintos sensores. Es necesario aclarar que los 32 GB disponibles de memoria serán distribuidos entre el sistema operativo, *software* necesario para realizar los distintos procesos de recolección y transmisión de datos, y el almacenamiento de datos.

Tabla II. **Especificaciones técnicas *Raspberry Pi 3 Modelo B+***

	<i>Raspberry Pi 3 modelo B+</i>
Procesador	Broadcom BCM2837B0, Cortex A53 (ARMv8)64-bit SoC.
Frecuencia de reloj	1,4 GHz.
Memoria	1 GB LPDDR2 SDRAM
Conectividad inalámbrica	2,4 GHz / 5GHz IEEE 802,11.b/g/n/ac Bluetooth 4,3, BLE.

Continuación de la tabla II.

Conectividad de Red	Gigabit Ethernet over USB 2,0 (300 Mbps de máximo teórico).
Puertos	GPIO 40 pines. HDMI 4 x USB 2,0 CSI (cámara <i>Raspberry Pi</i>) DSI (pantalla táctil) Tomas auriculares / video compuesto Micro SD Micro USB (alimentation) Power-over-Ethernet (PoE)
Fecha de lanzamiento	14 de marzo de 2018

Fuente: elaboración propia, empleado Microsoft Word 365.

2.2.4.5. Puertos GPIO de Raspberry Pi modelo 3 B+

General Purpose Input Output (GPIO) es un sistema de entrada y salida de propósito general, es decir, consta de una serie de pines o conexiones que se pueden usar como entradas o salidas para múltiples usos. Estos pines están incluidos en todos los modelos de *Raspberry Pi*, aunque entre modelos existen diferencias.

Entre sus características más destacadas:

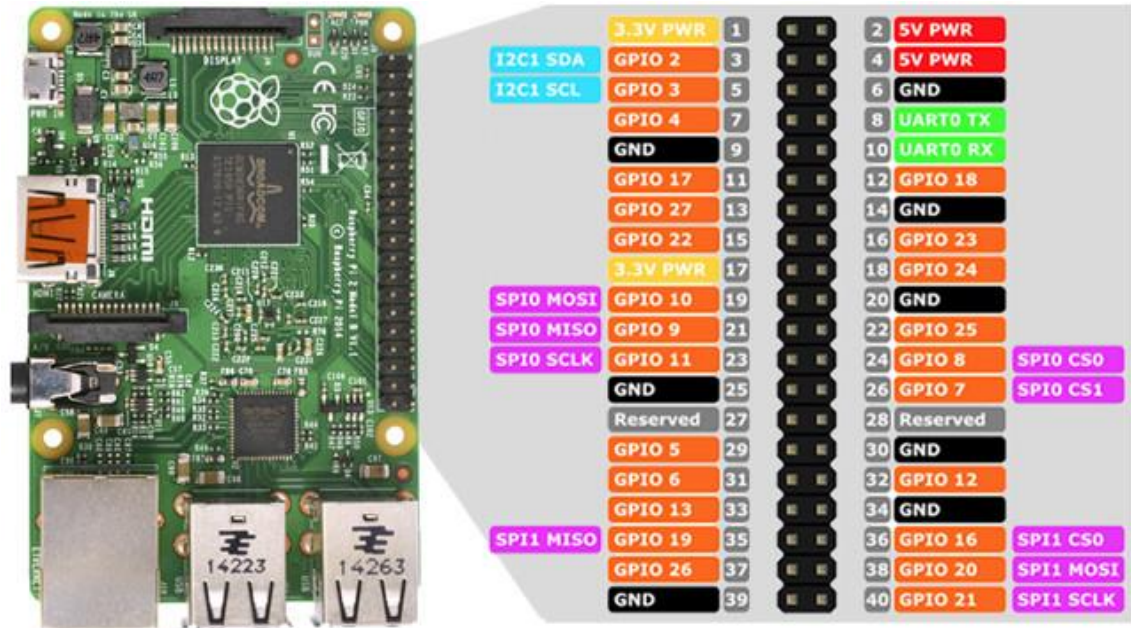
- Permiten ser configurados tanto como entrada como de salida. Tienen esa dualidad.
- Los pines GPIO pueden ser activados o desactivados mediante código.
- Permiten la lectura de datos binarios.

- Existen pines especiales que permiten regular el ancho del pulso (PWM).

En los GPIO se puede realizar la conectividad de dispositivos electrónicos como transistores, sensores de humedad/temperatura, termistores, motores paso a paso, LEDs, entre otros. También permite conectar componentes o módulos creados específicamente para *Raspberry Pi* y que extienden las capacidades de la placa más allá de lo que incluye de base. A estos se les conoce como hats, ya que se montan sobre la placa *Raspberry Pi*, coincidiendo con los GPIO necesarios para que puedan funcionar.

Por tanto, el montaje de los hats, es bastante simple y rápido; pero es necesario verificar la versión de compatibilidad de la placa, ya que existen diferencias entre los distintos modelos de *Raspberry Pi*.

Figura 18. Pines GPIO *Raspberry Pi* modelo 3 B+



Fuente: Raw. Pines GPIO *Raspberry Pi* modelo 3 B+.

<https://raw.githubusercontent.com/ProgramoErgoSum/Tutoriales/master/v1/introduccion-a-pines-gpio-en-raspbian/img/pines-gpio.jpg>. Consulta: 11 de octubre de 2021.

Una forma sencilla de ver el pinout de nuestra *Raspberry Pi* es introduciendo el comando `gpio readall`.

Figura 19. Ejemplo comando *gpio readall*

```

pi@raspberrypi:~$ 
pi@raspberrypi:~$ gpio readall
-----Pi 3-----
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name  | Mode | V  | Physical | V  | Mode | Name  | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2   | 8   | 3.3v  |      |    | 1         | 2   |      | 5v    |     |     |
| 3   | 9   | SDA.1 | IN   | 1  | 3         | 4   |      | 5v    |     |     |
| 4   | 7   | SCL.1 | IN   | 1  | 5         | 6   |      | 0v    |     |     |
|     |     | GPIO.7 | IN   | 1  | 7         | 8   | 1    | TXD   | 15  | 14  |
|     |     | 0v    |      |    | 9         | 10  | 1    | RXD   | 16  | 15  |
| 17  | 0   | GPIO.0 | IN   | 0  | 11        | 12  | 0    | GPIO.1 | 1   | 18  |
| 27  | 2   | GPIO.2 | IN   | 0  | 13        | 14  |      | 0v    |     |     |
| 22  | 3   | GPIO.3 | IN   | 0  | 15        | 16  | 0    | GPIO.4 | 4   | 23  |
|     |     | 3.3v  |      |    | 17        | 18  | 0    | GPIO.5 | 5   | 24  |
| 10  | 12  | MOSI  | IN   | 0  | 19        | 20  |      | 0v    |     |     |
| 9   | 13  | MISO  | IN   | 0  | 21        | 22  | 0    | GPIO.6 | 6   | 25  |
| 11  | 14  | SCLK  | IN   | 0  | 23        | 24  | 1    | CE0   | 10  | 8   |
|     |     | 0v    |      |    | 25        | 26  | 1    | CE1   | 11  | 7   |
| 0   | 30  | SDA.0 | IN   | 1  | 27        | 28  | 1    | SCL.0 | 31  | 1   |
| 5   | 21  | GPIO.21 | IN  | 1  | 29        | 30  |      | 0v    |     |     |
| 6   | 22  | GPIO.22 | IN  | 1  | 31        | 32  | 0    | GPIO.26 | 26  | 12  |
| 13  | 23  | GPIO.23 | IN  | 0  | 33        | 34  |      | 0v    |     |     |
| 19  | 24  | GPIO.24 | IN  | 0  | 35        | 36  | 0    | GPIO.27 | 27  | 16  |
| 26  | 25  | GPIO.25 | IN  | 0  | 37        | 38  | 0    | GPIO.28 | 28  | 20  |
|     |     | 0v    |      |    | 39        | 40  | 0    | GPIO.29 | 29  | 21  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name  | Mode | V  | Physical | V  | Mode | Name  | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
pi@raspberrypi:~$ 

```

Fuente: Circuits. *Comando gpio readall*. <https://www.circuits.dk/wp-content/uploads/2017/06/GPIO-readall-command-output.jpg> Consulta: 11 de octubre de 2021.

2.2.4.6. Arduino

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra o macho, dependiendo del modelo. Estos permites establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

Una placa electrónica es una PCB (*Printed Circuit Board*, Placa de Circuito Impreso). Son superficies planas fabricadas en un material no conductor, la cual consta de distintas capas de material conducto, el cual permite realizar las conexiones de señales análogas y digitales para su funcionamiento.

Una PCB es la forma más compacta y estable de construcción de un circuito electrónico. Por lo tanto, la placa Arduino, no es más que una PCB que

implementa un determinado diseño de circuitería interna. De esta forma, el usuario final no se debe preocupar por las conexiones eléctricas que necesita el microcontrolador para funcionar, y permite iniciar directamente el desarrollo de las distintas aplicaciones electrónicas que sean de interés.

Arduino es libre y extensible, así cualquiera que desee ampliar y mejorar el diseño de hardware de las placas como el entorno de desarrollo, pueden hacerlo sin problema. Esto permite que exista un rico ecosistema de placas electrónicas no oficiales para distintos propósitos y de librerías de *software* de tercero, que pueden adaptarse mejor a las necesidades de cada desarrollador.

Su entorno de programación es multiplataforma, se puede instalar y ejecutar en sistemas operativos como Windows, Mac OS y Linux.

Su lenguaje de programación es basado en C++, dicho lenguaje permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad tan grande, que los programadores más avanzados pueden expresar todo el potencial de su lenguaje y adaptarlo a cualquier situación.

2.2.4.7. *Arduino NANO*

El Arduino Nano es una placa de tamaño compacto, completa y compatible con *protoboards*, basada en el microcontrolador ATmega328P. Posee las mismas capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad, solo se ve recortado en su conector USB, conector Jack de alimentación y los pines cambia un formato de pines *header*.

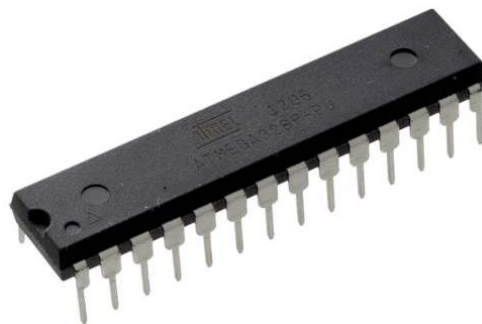
El microcontrolador ATMEGA328P es fabricado por Atmel (Microchip). Pertenece a la familia AVR con arquitectura RISC de 8 bits. Cuenta con poderosas instrucciones que se ejecutan en su mayoría en un solo ciclo de reloj, por lo que puede alcanzar un desempeño cercano a 1 MIPS por cada 1 MHz en la frecuencia de reloj.

Tabla III. **Características ATMEGA328P**

Microcontrolador ATMEGA328P	
Voltaje de operación	1,8 a 5,5 VDC.
Arquitectura de CPU	8 bit AVR.
Memoria flash	32 KB
Memoria RAM	2 KB
Frecuencia de operación	20MHz
Pines de IO	23
Canales ADC	10
Interfaces	UART, TWI, SPI.

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 20. **Microcontrolador ATMEGA328P**



Fuente: Geek Factory. Microcontrolador ATMEGA328P. <https://www.geekfactory.mx/wp-content/uploads/2016/11/atmega328p-microcontrolador-avr.jpg>. Consulta: 11 de octubre de 2021.

2.2.4.8. Puertos GPIO Arduino NANO

El microcontrolador tiene 8 pines analógicos con un convertidor ADC de 10 bits. El ADC convierte el voltaje de entrada en una señal entre 0 y 1 023. También es posible utilizar los pines analógicos A0 – A5 como salida con la función *digitalWrite* (Ax). Los dos pines analógicos A6 y A7 son solo entradas analógicas.

Hay un total de 14 pines de E/S digitales que se pueden usar con una moneda máxima de 40 mA si no está usando los puertos serie RX0 (UART In) y TX1 (UART Out). En la tabla III se muestra a detalle los catorce pines de entradas y salidas de la placa Arduino NANO.

Tabla IV. Características GPIO Arduino NANO

Categoría pin	Nombre de pin	Detalle
Poder	Vin, 3,3 V, 5 V, tierra	Vin: voltaje de entrada a Arduino cuando se usa una fuente de alimentación externa (6 – 12V). 5V: fuente de alimentación regulada que se utiliza para alimentar el microcontrolador y otros componentes de la placa. 3,3V: suministro de 3,3V generado por el regulador de voltaje incorporado. GND: clavijas de tierra.
Reiniciar	RESET	Reiniciar el microcontrolador.
Pines analógicos	A0 – A7	Se utiliza para medir voltaje analógico en el rango de 0 – 5 V.
Pines de entrada/salida.	Pines digitales D0 – D13	Se puede utilizar como pines de entrada o salida 0V (bajo) y 5V (alto).
Serie	Rx, Tx	Se utiliza para recibir y transmitir datos en serie en TTL
Interrupciones externas	2, 3	Para desencadenar una interrupción.

Continuación de la tabla IV.

PWM	3, 5, 6, 9, 11	Proporciona salidas PWM de 8 bits.
SPI	10 (SS), 11 (MOSI), 12 (MISO) y 13 (SCK)	Se utiliza para la comunicación SPI.
LED incorporado	13	Para encender el LED incorporado
IIC	A4 (SDA), A5 (SCA)	Se utiliza para la comunicación TWI.
AREF	AREF	Para proporcionar un voltaje de referencia para el voltaje de entrada.

Fuente: elaboración propia, empleando Microsoft Word 365.

3. **SENSORES QUE COMPONEN LA ESTACIÓN METEOROLÓGIA DE BAJO COSTO IMPRESO EN 3D.**

3.1. **Sensor de presión BMP280**

El sensor de presión barométrica BMP280 permite medir la altura respecto al nivel del mar, su funcionamiento está basado en la relación entre presión del aire y la altitud.

Este sensor posee alta precisión y bajo consumo de energía. Ofrece un rango de medición de 300 a 1 100 hPa. Ha sido diseñado para ser conectado directamente a un microcontrolador a través de I2C o SPI.

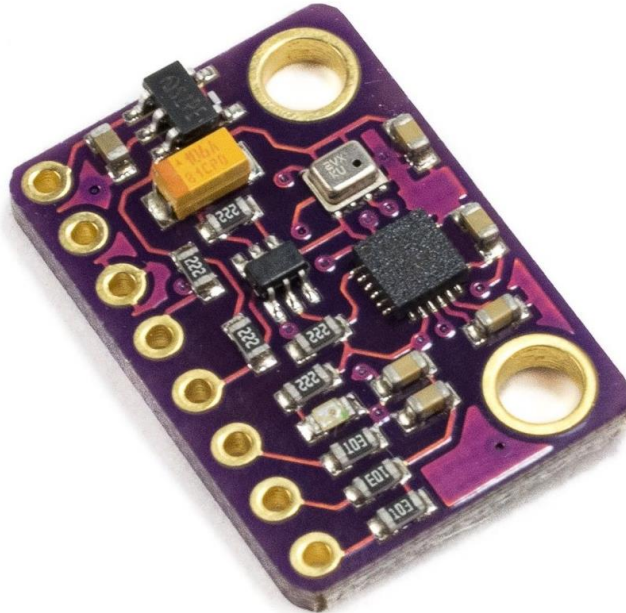
Este tipo de sensores pueden ser utilizados para calcular la altitud con gran precisión, por lo que es un sensor muy utilizado en sistemas de Autopiloto para Drones (UAVs). Este sensor puede entregar medidas de altitud con una precisión de hasta 1m.

Tabla V. **Especificaciones técnicas sensor BMP280**

Rango de presión	300 ... 1 100 hPa.
Coeficiente de temperatura de salida	1,5 Pa/K, equiv. A 12,6 cm/K (25 ... 40 °C @900hPa).
Interfaces digitales	I2C (arriba de 2,4MHz). SPI (arriba de 10MHz).
Rango de temperatura en operación	-40 ... +85 °C

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 21. **Sensor de presión BMP280**



Fuente: Naylamp Mechatronics. *Sensor de presión BMP280*.

https://naylampmechatronics.com/1481-superlarge_default/modulo-gy-91-mpu9250-bmp280-acelerometro-giroscopio-magnetometro-altimetro-i2c.jpg. Consulta: 11 de octubre de 2021.

Tabla VI. **Descripción de pines BMP280**

Nombre	Descripción
GND	Tierra
CSB	Selección de Chip
SDI	Entrada serial de datos
SCK	Entrada serial de reloj
SDO	Salida serial de datos
V _{DDIO}	Interfaz digital de alimentación
GND	Tierra
V _{DD}	Alimentación analógica

Fuente: elaboración propia, empleando Microsoft Word 365.

3.2. Sensor de temperatura MCP9808

Es un sensor de temperatura digital controlado por un puerto I2C. Este sensor es programable según el usuario para obtener valores de temperatura, puede ser seleccionado según su configuración como apagado o valores bajos de poder; y también permite obtener una ventana de alerta según los límites establecidos en los valores críticos de salida. Todo esto depende según la polaridad seteada en el sensor. Este sensor trabaja a una frecuencia de 400kHz y es compatible con interfaz serial.

Tabla VII. **Especificaciones técnicas sensor MCP9808**

Exactitud	$\pm 0,5 \text{ }^{\circ}\text{C}$
Resolución	$+ 0,0625 \text{ }^{\circ}\text{C}$
Bits ADC	12 bits
Comunicación	I2C
Rango de medición	- 20°C hasta $+ 100^{\circ}\text{C}$
Consumo	200 μA

Fuente: elaboración propia, empleando Microsoft Word 365.

Tabla VIII. **Descripción pines MCP9808**

Nombre	Descripción
SDA	Línea serial de datos
SCL	Línea serial del reloj
GND	Tierra
V _{DD}	Pin de alimentación

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 22. **Sensor MCP9808**



Fuente: Bing. *Sensor MCP9808*.

<https://www.bing.com/images/search?q=Sensor%20MCP9808&qs=n&form=QBIR&sp=1&pq=sensor%20mcp9808&sc=0->

14&cvid=51225BE47422421B849AEEAF53D1FE97&first=1&tsc=ImageHoverTitle. Consulta: 11 de octubre de 2021.

3.3. **Sensor de humedad HTU21D**

Es un sensor de humedad digital con salida de temperatura, también digital. Entre sus características principales provee calibración directamente de fábrica, señales digitales de liberalización y trabaja mediante I2C.

Este sensor posee un sensor aislado en una capsula de plástico para mantenerlo limpio, un regulador de 3,3V y un conversor de nivel lógico para la comunicación I2C. Esto permite ser utilizado de forma segura con cualquier tipo de microcontrolador con potencia de 3,3V-5V.

El HTU21D posee una precisión típica de $\pm 2\%$ con un rango de operación de 5 % a 95 % de Humedad Relativa. El funcionamiento fuera de los rangos especificados no es preciso en su medición.

La temperatura tiene una precisión de ± 1 °C y rango de -30 ~ 90 °C.

Tabla IX. **Descripción de pines HTU21D**

Nombre	Descripción
DATA	Bit de datos
GND	Tierra
NC	No se conecta
NC	No se conecta
VDD	Alimentación
SCK	Selector para Humedad Relativa o Temperatura

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 23. **Sensor de humedad HTU21D**



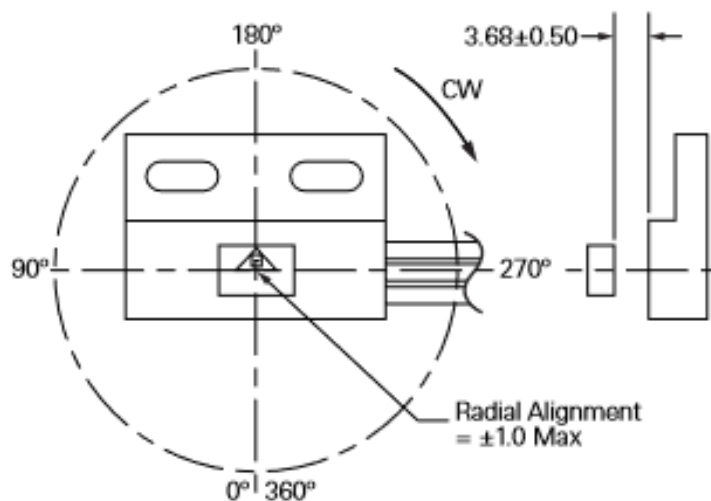
Fuente: ML Static. *Sensor de humedad*. https://http2.mlstatic.com/D_NQ_NP_708242-MLM42673146769_072020-O.webp. Consulta: 11 de octubre de 2021.

Sensor de dirección de viento *Littelfuse 55300-00-002-A Flat Pack Sensor* Rotativo

Los sensores de efecto Hall son sensores magnéticos cuya salida depende del campo magnético o de la densidad de flujo magnético alrededor del sensor magnético.

La serie *Littelfuse 55 300* de sensores rotativos de posición absoluta proporciona medida entre 0 y 360 grados. El diseño interno del sensor utiliza tecnología de efecto Hall magnético sin contacto con opciones de señales analógicas de salida PWM. Una ventaja principal de este sensor que no existe desgaste mecánico mediante el uso.

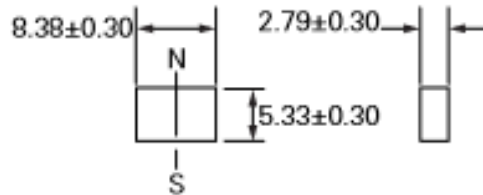
Figura 24. **Actuador magnético de rotación**



Fuente: Mouser. *Actuador Magnético de Rotación*.

[https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.p](https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.pdf-938902.pdf)
df-938902.pdf Consulta: 11 de octubre de 2021.

Figura 25. **Actuador magnético**

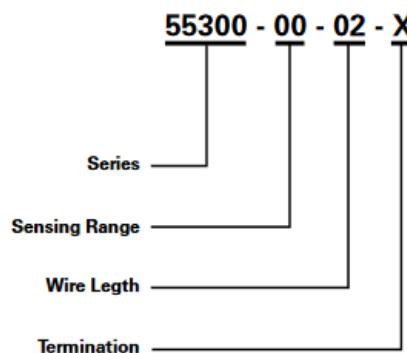


Fuente: Mouser. *Actuador magnético*.

https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.pdf
df-938902.pdf. Consulta: 11 de octubre de 2021.

En la figura 24 se observa la posición que debe tener el sensor *Littelfuse* 55300 para obtener una coherencia en los datos al momento de ser implementado. En la figura 25 se describen las cotas y polaridad del actuador magnético para optimizar el funcionamiento del sensor *Littelfuse*.

Figura 26. **Sistema de numeración de piezas sensores *Littelfuse***



Fuente: Mouser. *Littelfuse Hall Effect Sensors 55300 Datasheet*.

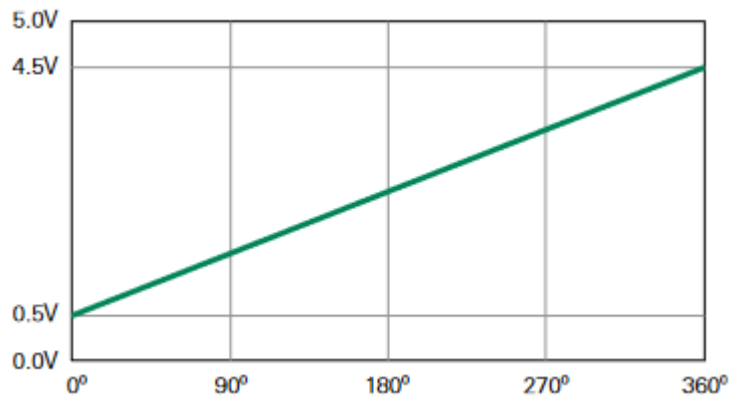
https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.pdf
df-938902.pdf Consulta: 11 de octubre de 2021.

Tabla X. **Opciones de terminación sensores *Littelfuse***

Opción a seleccionar	Descripción	Colores	Función
A	Cables pelados	Negro Blanco Rojo	Tierra Señal V(in)
C	Conector Molex MX 150 33481-0301	Negro Blanco Rojo	Pin 1: Negro/Tierra Pin 2: Blanco/Señal Pin 3: Rojo/V(in)

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 27. **Señal de salida**



Fuente: Mouser. *Littelfuse Hall Effect Sensors 55300 Datasheet*.

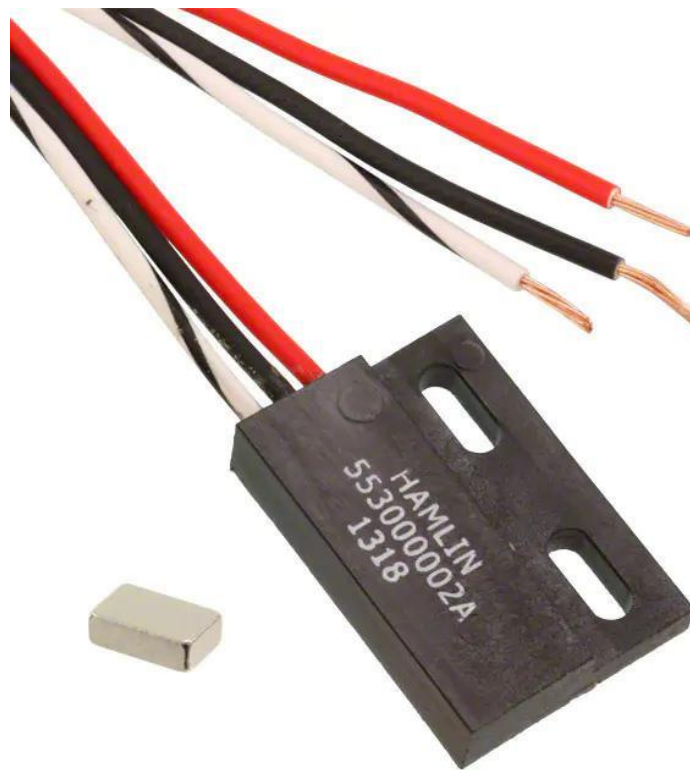
[https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.p
df-938902.pdf](https://www.mouser.com.gt/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55300_Datasheet.pdf-938902.pdf). Consulta: 11 de octubre de 2021.

En la figura 27 se ejemplifica la nomenclatura de los sensores *Littelfuse*, y en la tabla IX se describe las dos opciones dadas por el fabricante, en donde su única diferencia es la implementación del conector Molex MX 150 33 481-0301 al sensor con terminación C en comparación al sensor con terminación A.

Para este proyecto se utilizó el sensor *Littelfuse* 55300-00-02-A por conveniencia.

En la figura 28 se representa gráficamente la relación directamente proporcional de la señal analógica como entrada al valor de salida representada en grados, teniendo como valor mínimo 0,5 V equivalente a 0°, y valor máximo 4,5V equivalente a 360 °.

Figura 28. **Sensor 55 300-00-02-A**



Fuente: Amazon S3. *Sensor 55 300-00-02-A*. <https://s3-eu-west-1.amazonaws.com/enrgtech-images/full/f7a3b7df276ef8b2d8fd66497a384b6dba154d77.jpg> Consulta: 11 de octubre de 2021.

Sensor de velocidad de viento y precipitación Honeywell SS451 A bipolar por efecto Hall

El sensor SS451A es un dispositivo de efecto Hall con señales digitales y versátil, que es operado por el campo magnético de un imán permanente o un electroimán. Está diseñado para responder a un polo norte o un polo sur.

Este tipo de sensores omnipolares, son dispositivos sensibles y flexibles diseñados para satisfacer una amplia gama de aplicaciones potenciales. Para este proyecto son utilizados para recolectar datos de velocidad de viento y precipitación. Debido a que pueden ser operados por un polo norte o un polo sur, no requieren que se identifique la polaridad del imán, lo que facilita la instalación y reduce potencialmente el costo del sistema. Sus características principales son:

- Activación simple desde un polo norte o un polo sur, y magnetismo sensible.
- Capacidad de 3Vcc de bajo voltaje.
- Protección de polaridad inversa incorporada.
- Circuito integrado térmicamente equilibrado.

Para este tipo de circuitos integrados que componen el sensor SS451A, permite tener una salida inicial en el estado encendido o apagado, si se enciende con un imán aplicado generando un campo magnético.

Para estabilizar la salida de la señal generada por el imán, es necesario únicamente 3V. Este sensor consta únicamente de tres pines, los cuales son suministro de voltaje, tierra y el pin de salida.

Figura 29. **Sensor SS451A**



Fuente: Media Digikey. *Sensor SS451A*.

<https://media.digikey.com/Photos/Honeywell%20Photos/SS451A.JPG>. Consulta: 11 de octubre de 2021.

3.4. Sensor de intensidad lumínica SI1145

Este sensor permite calcular el índice UV. No contiene un sensor de UV real, sino que lo aproxima en función de la luz visible e IR del sol. Es un sensor digital que funciona mediante interfaz digital I2C, el cual incluye un sensor ADC, fotodiodos visibles e infrarrojo de alta sensibilidad. El sensor posee elementos de detección de infrarrojos y visibles individuales para que se pueda medir casi cualquier tipo de luz. Trabaja mediante proximidad infrarroja de baja potencia, basada en reflectancia.

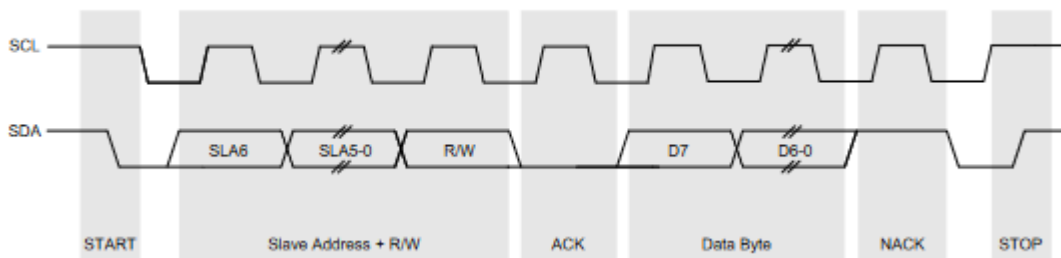
Los circuitos de conversión digital brindan una excelente inmunidad al ruido de parpadeo en las luces artificiales y ruido de aleteo de luz natural. En los pines del sensor SI1145 posee tres esenciales, los cuales son:

- SCL.
- SDA.

- INT.
- SCL y SDA son clavijas de drenaje abierto estándar que se requieren para la operación I2C. El pin INT interrumpe el proceso realizado. Se necesita una resistencia para un funcionamiento correcto. Como salida de drenaje abierto, el pin INT, se puede compartir con otras interrupciones en el sistema.

El reinicio de transmisión I2C debe enviarse antes de cualquier acceso de registro I2C al sensor SI1145. Si cualquier registro o parámetro I2C ya se ha escrito en el sensor, cuando se emite el reinicio de transmisión, el host debe enviar un comando de reinicio por completo.

Figura 30. **Diagrama de sincronización de bits I2C**



Fuente: CDN Shop. *Si1145/46/47*. <https://cdn-shop.adafruit.com/datasheets/Si1145-46-47.pdf>.

Consulta: 12 de octubre de 2021.

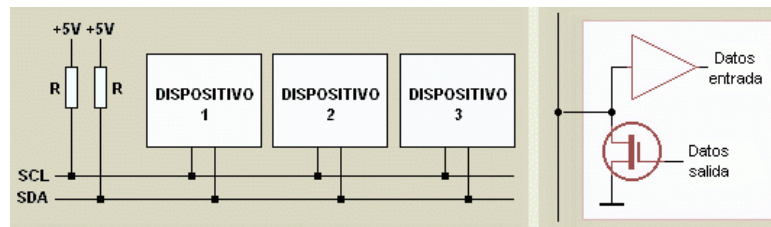
La condición inicial, de bus libre, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de inicio. Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en estado alto la línea de reloj (SCL).

El primer byte que se transmite luego de la condición de inicio siete bits, los cuales pueden ser apreciados en la figura 31, que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

Los datos se transfieren en secuencias de 8 bits. Estos bits se colocan en la línea SDA comenzando por el bit de más peso (más significativo). Una vez puesto un bit en SDA, se lleva la línea SCL a alto.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectado a la alimentación por medio de resistores “pull-up”) lo que define a una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

Figura 31. Diagrama de comunicación I2C



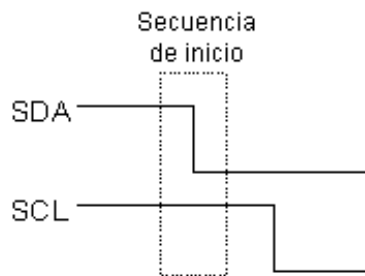
Fuente: Robots Argentina. *Descripción y funcionamiento del bus i2c*. <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>. Consulta: 12 de octubre de 2021.

Las dos líneas del bus están en un nivel lógico alto cuando están inactivas. En principio, el número de dispositivos que se pueden conectar al bus no tiene límites, aunque hay que observar que la capacidad máxima sumada de todos los dispositivos no supere los 400 pF. El valor de los resistores de polarización no es

muy crítico, y puede ir desde 1,8 KΩ a 47 KΩ. Un valor menor de resistencia incrementa el consumo de los integrados, pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales.

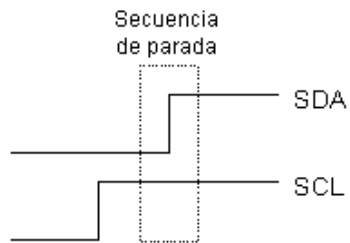
Para este proyecto, es necesario tomar en cuenta que, habiendo varios dispositivos conectados sobre el bus, es lógico que para establecer una comunicación a través de él se deba respetar un protocolo.

Figura 32. **Diagrama de secuencia de inicio protocolo I2C**



Fuente: Robots Argentina. *Descripción y funcionamiento del bus i2c*. <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/> Consulta: 12 de octubre de 2021.

Figura 33. **Diagrama de secuencia de finalización protocolo I2C**



Fuente: Robots Argentina. *Descripción y funcionamiento del bus i2c*. <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>. Consulta: 12 de octubre de 2021.

4. DISEÑO DE UNA ESTACIÓN METEOROLÓGICA IMPRESA EN 3D

Con los requerimientos a cubrir para el diseño e implementación de telemetría para una estación meteorológica en los capítulos anteriores, así como los diferentes componentes a utilizar en la recolección de las distintas variables meteorológicas a registrar, este capítulo está orientado a explicar el diseño del producto final; así como el *software* utilizado para realizar la adquisición, procesamiento y presentación de datos mediante los distintos sensores y el módulo de transmisión.

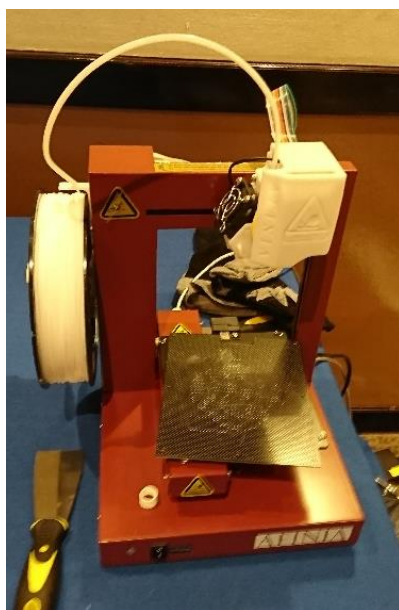
Para ello se utilizaron los diseños propuestos por la Corporación Universitaria para la Investigación Atmosférica (University Corporation for Atmospheric Research, UCAR) y la Oficina de Asuntos Internacionales del Servicio Meteorológico Nacional (National Weather Service International Activities Office, NWS IAO) de los EE.UU. Descargados directamente de su sitio web oficial, el cual puede ser consultado en la sección de bibliografías de este proyecto.

Los archivos de impresión 3D se encuentran en formato STL, siendo el formato de transmisión de datos estándar por excelencia de la industria de creación rápida de prototipos.

Los modelos de impresoras 3D utilizados fueron Afinia H480, para las piezas pequeñas, y Afinia H800. Equipo también denominado manufactura por adición, ya que permite producir objetos a través de la adición de material en capas que corresponden a las sucesivas secciones transversales de un modelo

3D, para las piezas grandes; las cuales fueron proporcionadas por INSIVUMEH; al igual que el filamento ASA (plástico que mantiene su aspecto y resistencia al impacto después de estar expuesto a largos intervalos de tiempo al aire libre, lluvia, frío y al agua salada del mar).

Figura 34. **Impresora 3D Afinia H480**



Fuente: elaboración propia. INSIVUMEH.

Figura 35. **Impresora 3D Afinia H800**



Fuente: elaboración propia. INSIVUMEH.

Diseño de recubrimiento, impreso en 3D, para los sensores de la estación meteorológica

4.1.1. *Conjunto del protector de radiación para los sensores BMP280, MCP9808 y HTU21D*

Este protector de radiación cumple la función de amparar los sensores de temperatura, humedad y presión de la exposición directa a la radiación solar y permitir una ventilación adecuada para que el aire muestreado sea representativo de las condiciones al aire abierto.

La lista de piezas para su elaboración se lista en la tabla X.

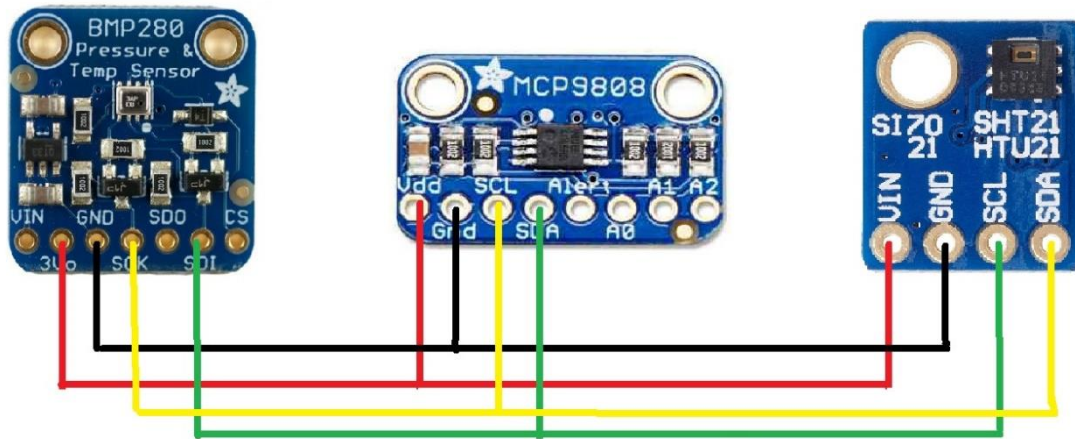
Tabla XI. **Lista de piezas para protector de radiación para sensores BMP280, MCP9808 y HTU21D**

Descripción	Unidad	Cantidad
Plato superior (impreso en 3D)	Pieza	1
Plato cerrado (impreso en 3D)	Pieza	1
Plato abierto (impreso en 3D)	Pieza	5
Varilla roscada (impreso en 3D)	Pieza	1
Disco perforado central (impreso en 3D)	Pieza	1
Disco perforado abierto (impreso en 3D)	Pieza	4
Base (impreso en 3D)	Pieza	1
Tornillo (impreso en 3D)	Pieza	2
Rosca de enclavamiento (impreso en 3D)	Pieza	1
Tuerca de fijación (impresa en 3D)	Pieza	1

Fuente: elaboración propia, empleando Microsoft Word 365.

Se realizó la interconexión de los sensores BPM280, MCP9808 y HTU21D, mediante un sistema de integración, ya que estos sensores son ubicados en la misma estructura que los protege y funcionan mediante el protocolo I2C.

Figura 36. **Esquema de conexión sensores BMP280, MCP9808 y HTU21D**



Fuente: elaboración propia, empleando Microsoft PowerPoint 365.

Con el objetivo de reducir costos y simplificar el proceso de cableado, se realizó conector común de 4 pines para distribuir las conexiones de corriente, tierra común y señales a los distintos sensores, siendo descritos las piezas utilizadas en la siguiente tabla.

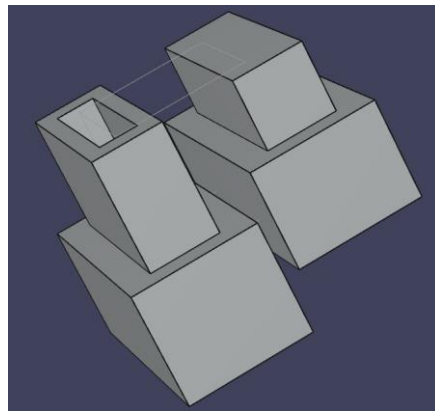
Tabla XII. **Piezas conector común**

Descripción	Unidad	Cantidad
Bloque de 4 pines	Pieza	4
Cable puente hembra a hembra 10 cm.	Pieza	12
Cable puente macho a hembra de 20 cm	Pieza	4
Conector común (base y tapa) de 4 pines (impreso 3D)	Pieza	4

Fuente: elaboración propia, empleando Microsoft Word 365.

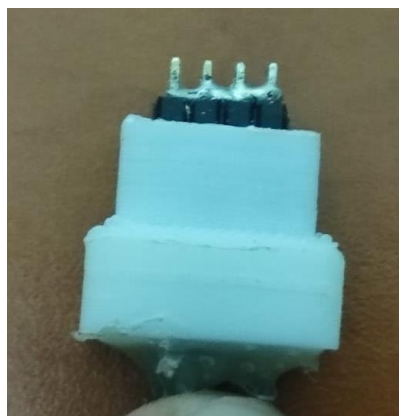
Los bloques de cuatro pines fueron soldados para generar el bloque común de conexión en cada uno de los bloques impresos, las tapas son utilizadas para resguardar el área descubierta y proteger las conexiones de los sensores que puedan generar un corto circuito por lluvia.

Figura 37. **Vista 3D conectores comunes**



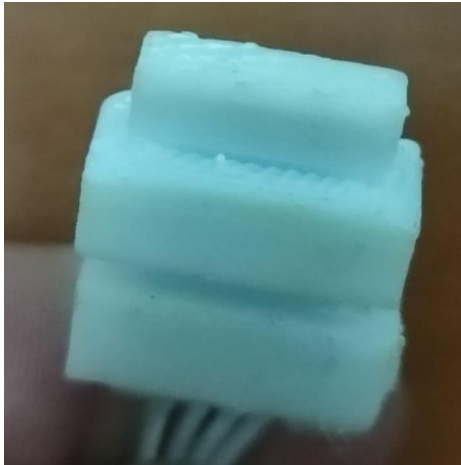
Fuente: elaboración propia, empleando Robot Structural 365.

Figura 38. **Impresión 3D conector común base y bloque de pines**



Fuente: elaboración propia. INSIVUMEH.

Figura 39. **impresión 3D conector común base y tapa**



Fuente: elaboración propia. INSIVUMEH.

En la figura 38 se observa el diseño en 3D de los conectores comunes utilizados, en las figuras 39 se observa el resultado final de la base y la utilización de los bloques de pines utilizados. En la figura 40 se muestra el resultado final del conector común con su base y tapa de protección.

4.1.2. *Conjunto de acople para recolección de datos y protección de sensor SS541A, sensor de velocidad de viento*

El anemómetro de tres cazoletas de la estación meteorológica emplea un sensor de enganche de efecto Hall activado por dos pequeños imanes para contar las revoluciones de las cazoletas. La *Raspberry Pi* convierte la cantidad de revoluciones convirtiéndola en un valor de velocidad lineal.

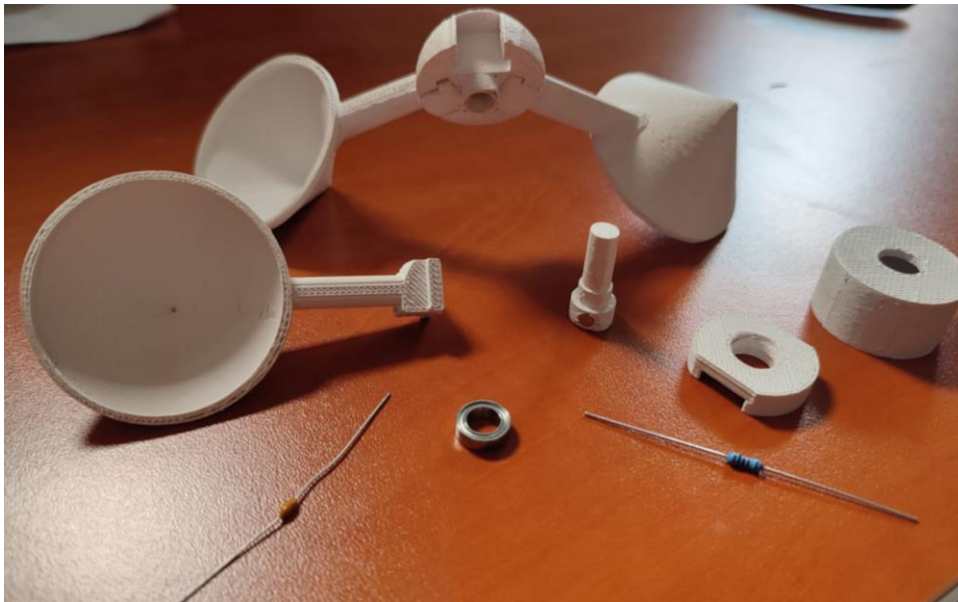
Los materiales utilizados para la elaboración y acople al sistema de describen en la tabla XII.

Tabla XIII. **Lista de piezas utilizadas en anemómetro**

Descripción	Unidad	Cantidad
Cazoleta (impresa en 3D).	Pieza	3
Cubo (impresa en 3D).	Pieza	1
Portarrodamiento (impreso en 3D).	Pieza	1
Eje conductor	Pieza	1
Imán de neodimio	Pieza	2
Porta sensor	Pieza	1
Sensor Honeywell SS451A bipolar de efecto Hall).	Pieza	1
Condensador (0,1 microfaradios)	Pieza	1
Resistencia (1000 ohmios)	Pieza	1
Cojinete [10 mm (d.e.) x 6 mm (d.i.) x 3 mm (espesor)]	Pieza	1
Cable puente macho a macho de 10 cm	Pieza	3

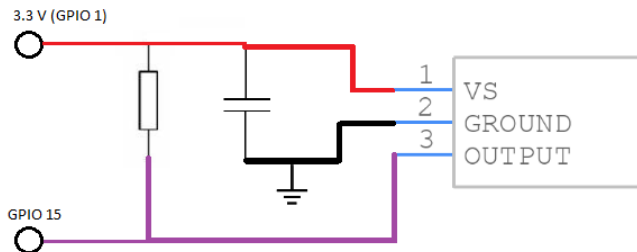
Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 40. **Cazoleta y equipo para anemómetro**



Fuente: elaboración propia. INSIVUMEH.

Figura 41. Diagrama de conexión sensor SS451



Fuente: elaboración propia, empleando Microsoft PowerPoint 365.

4.1.3. Conjunto de acople para recolección de datos y protección del sensor *Littelfuse 55300-00-02-A*

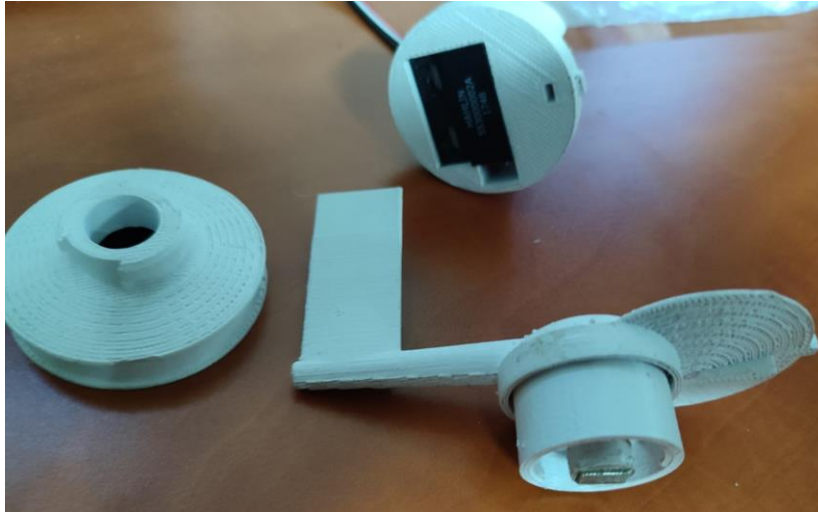
Esta estación utiliza un sensor giratorio de efecto Hall para la recolección de datos de la dirección de viento, empleando el sensor *Littelfuse 55300-00-02-A*.

Tabla XIV. Lista de piezas utilizadas en anemómetro

Descripción	Unidad	Cantidad
Rotor (impreso en 3D)	Pieza	1
Cuerpo (impreso en 3D)	Pieza	1
Eje conductor (impreso en 3D)	Pieza	1
Cubierta (impresa en 3D)	Pieza	1
Collar de orientación y abrazadera (impreso en 3D)	Pieza	1
Perno de collar de orientación	Pieza	1
Imán	Pieza	1
Sensor giratorio en paquete plano <i>Littelfuse 55 300-00-02-A</i>	Pieza	1
Cojinete [10 mm (d.e.) x 6 mm (d.i.) x 3 mm (espesor)]	Pieza	1

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 42. **Veleta y sensor *Littelfuse SS300-00-02-A***



Fuente: elaboración propia. INSIVUMEH.

En la figura 43 se logra apreciar, la utilización de un imán de neodimio en la parte inferior de la veleta, este interactúa directamente con el sensor SS300-00-02-A ubicado en la parte de la base de la estructura impresa en 3D. Este imán fue acoplado mediante una lima para poder introducirlo directamente en la estructura y se utilizó pegamento para su fijación.

4.1.4. *Conjunto de acople para recolección de datos y protección del sensor Honeywell SS451A, sensor de precipitación*

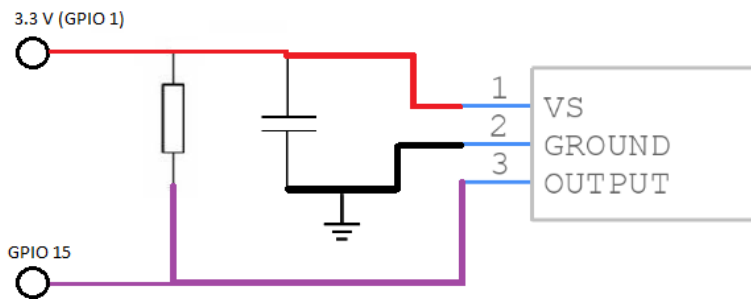
El pluviómetro de cubeta basculante estándar emplea la utilización de un sensor de enganche de efecto Hall activado por un pequeño imán, el cual permite contabilizar las veces que la cubeta se vuelca. Los materiales para utilizar para la protección del sensor, acoplado a la recolección de la variable de precipitación se listan en la tabla XIV.

Tabla XV. **Lista de piezas utilizadas en el pluviómetro**

Descripción	Unidad	Cantidad
Cono superior (impreso en 3D)	Pieza	1
Embudo (impreso en 3D).	Pieza	1
Cubeta (impresa en 3D)	Pieza	1
Remache pop (de clavo) de 3,5 mm	Pieza	2
Imán	Pieza	1
Topes (impreso en 3D)	Pieza	2
Condensador (0,1 microfaradio)	Pieza	1
Cables puente macho a macho de 10 cm	Pieza	3
Resistencia (1 000 ohmios)	Pieza	1
Sensor de efecto Hall SS451	Pieza	1
Porta sensor	Pieza	1
Cubierta (impresa en 3D)	Pieza	1

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 43. **Diagrama de conexión sensor SS451**



Fuente: elaboración propia, empleando Microsoft PowerPoint 365.

La resolución estándar de un pluviómetro es de 0,2 mm por volcado. Para asegurarse de que el pluviómetro fabricado cumpla con las especificaciones, este fue sometido a calibración, realizando mediciones comparativas con una estación automática instalada en las instalaciones de INSIVUMEH.

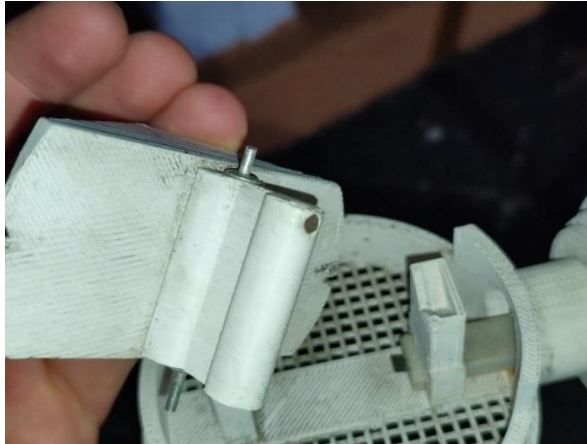
Por es una estación hecha mediante plástico ASA, este material permite realizar desgaste de las piezas para conseguir corregir errores empleados en el momento de impresión; este desgaste fue realizado en los topes que se encuentran acoplados en los laterales de la base del pluviómetro.

Figura 44. **Topes desgastados para calibración de toma de datos sobre precipitación**



Fuente: elaboración propia. INSIVUMEH.

Figura 45. **Cubeta o balancín para toma de muestras de precipitación**



Fuente: elaboración propia. INSIVUMEH.

Figura 46. **Cono superior y embudo**



Fuente: elaboración propia. INSIVUMEH.

En la figura 45 se aprecia el desgaste realizado a los topes para la calibración y coincidir con valores que se registren en la estación meteorológica impresa en 3D. En la figura 46 se observa la utilización de los remaches, el cual permiten realizar el volteado al momento de superar los 0,2 mm de acumulado de lluvia, y en la parte posterior se verifica la implementación de un imán para interactuar con el sensor SS451A. En la figura 47 se muestra el resultado final del pluviómetro.

4.1.5. **Conjunto de protección para el sensor de intensidad lumínica SI1145**

Este sensor mide los niveles de radiación en los espectros ultravioleta (UV), infrarrojo (IR) y visible, utilizando el sensor *Adafruit* SI1145.

En la tabla XVI se describen los materiales utilizados para su elaboración.

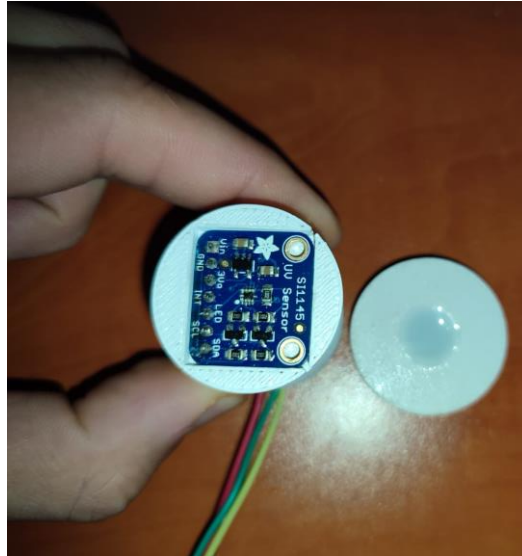
Tabla XVI. **Listas de piezas utilizadas en el sensor de radiación solar.**

Descripción	Unidad	Cantidad
Cubierta (impresa en 3D)	Pieza	1
Base (impresa en 3D)	Pieza	1
Cable puente macho a hembra de 10 cm.	Pieza	4
Disco de PTFE (politetrafluoroetileno) de 1,5 mm de espesor	Pieza	1
Sensor <i>Adafruit</i> SI1145	Pieza	1

Fuente: elaboración propia, empleando Microsoft Word 365.

En la figura 48 se observa el ensamblaje final del conjunto protector del sensor lumínico SI1145, y en la figura 49 se observa el ajuste del PTFE acoplado a la tapa, siendo pegada cuidadosamente para no afectar el área de interés al momento de obtener valores de intensidad lumínica.

Figura 47. **Sensor y soporte para sensor SI1145**



Fuente: elaboración propia. INSIVUMEH.

Figura 48. **Fijación de PTFE a tapa**



Fuente: elaboración propia. INSIVUMEH.

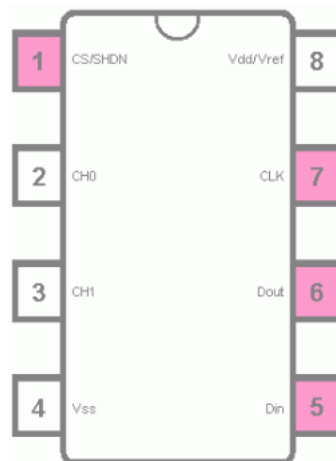
4.2. Conversor ADC

Es un sistema que transforma señales analógicas en señales digitales. Las señales para transformar es tensión o corriente, mediante un cuantificador y codificando las señales en código binario.

4.2.1. MCP3002

El MCP3002 es un circuito integrado (IC) convertidor analógico a digital (ADC), que interactúa con interfaz periférica en serie (SPI) de la *Raspberry Pi*. Este IC utiliza cuatro pines para la interfaz SPI, como se muestra en la figura 50. Tiene una resolución de 10 bits, que es lo suficientemente óptimo para una amplia gama de aplicaciones como sensores de luz y temperatura. Este IC es de baja potencia que funciona con un voltaje de 2,7 V.

Figura 49. **Raspberry Pi ADC, MCP3002**



Fuente: Petervis. *Raspberry Pi Analog to Digital Converter*.

https://www.petervis.com/Raspberry_PI/Analog_to_Digital_Converter/Raspberry_Pi_Analog_to_Digital_Converter.html Consulta: 13 de octubre de 2021.

Tabla XVII. **Conexión de pines SPI**

Pines SPI del IC MCP3002	Pines SPI de <i>Raspberry Pi</i>
1	24 o 26
5	19
6	21
7	23

Fuente: elaboración propia, empleando Microsoft Word 365.

La tabla de conexiones anteriores muestra los pines SPI en la *Raspberry Pi* GPIO y su conexión a los pines SPI en el IC MCP2003. Por lo general, la interfaz SPI de este circuito integrado funciona en modo esclavo, mientras que la interfaz SPI en la *Raspberry Pi* funciona en modo maestro.

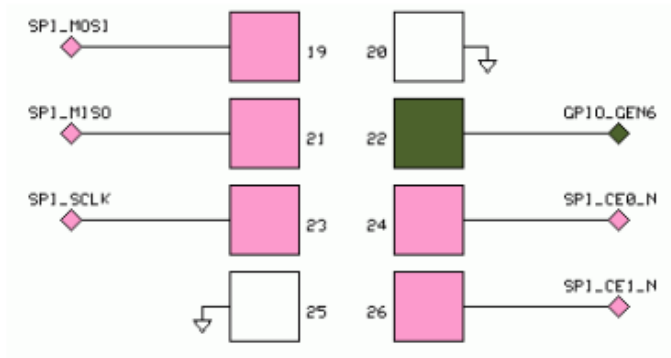
La interfaz SPI de *Raspberry Pi* puede comunicarse con hasta dos dispositivos esclavos, lo realiza eligiendo el dispositivo esclavo a través del 24 de selección de chip y el pin 26. Por lo tanto, el pin 1 del MCP2003 puede conectarse al pin 24 o al pin 26.

El pin 5 en el MCP2003 es la entrada de datos, que se conecta en el GPIO 19 de la *Raspberry Pi*, que es *Master Out Slave In* – MOSI.

El pin 6 en el MPC2003 es la salida de datos, el cual va conectado al GPIO 21 de *Raspberry Pi*, que es *Master In Slave Out* – MISO.

El pin 7 en el MCP2003 requiere una señal de reloj proporcionada por el GPIO 23 de la *Raspberry Pi*.

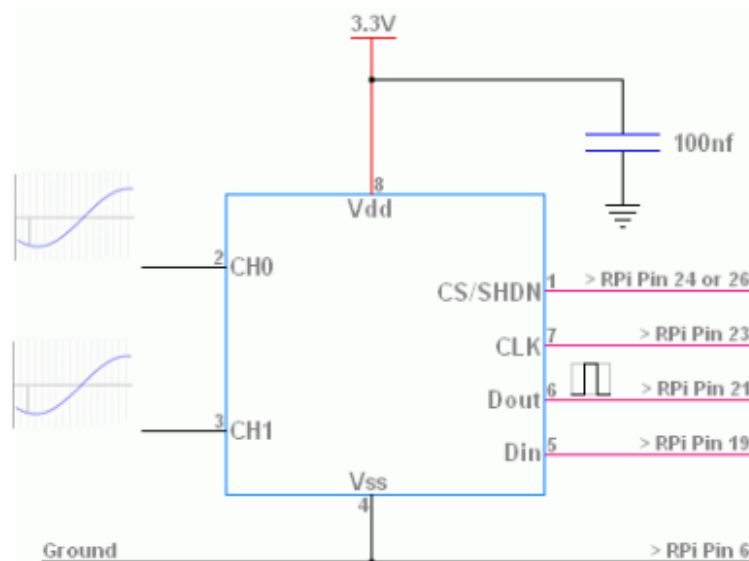
Figura 50. **Conexión de los pines SPI**



Fuente: Petervis. *Raspberry Pi Analog to Digital Converter*.

https://www.petervis.com/Raspberry_PI/Analog_to_Digital_Converter/Raspberry_Pi_Analog_to_Digital_Converter.html. Consulta: 13 de octubre de 2021.

Figura 51. **Diagrama del circuito del MCP2003**



Fuente: Petervis. *Raspberry Pi Analog to Digital Converter*.

https://www.petervis.com/Raspberry_PI/Analog_to_Digital_Converter/Raspberry_Pi_Analog_to_Digital_Converter.html. Consulta: 13 de octubre de 2021.

Como se puede observar en la figura 52, los canales CH1 Y CH0 pueden muestrear valores analógicos. El IC convierte este valor en digital y lo transfiere a la *Raspberry Pi* a través del SPI que consta de los pines 1, 5, 6 y 7.

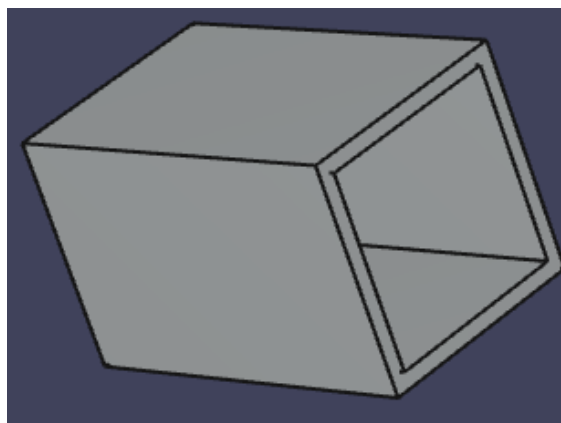
En la implementación de este circuito integrado, y asegurar su conexión segura a la *Raspberry Pi* se utilizaron los siguientes materiales, listados en la tabla XVIII.

Tabla XVIII. **Piezas de ensamble circuito integrado MCP2003**

Descripción	Unidad	Cantidad
Cubierta (impresa en 3D)	Pieza	1
Convertidor ADC MCP3002	Pieza	1
Placa de circuitos perforada con orificios metalizados (PTH)	Pieza	1
Cable puente macho a hembra de 20 cm	Pieza	6
Cable puente macho a macho de 10 cm	Pies	2

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 52. **Diseño en 3D de la cubierta del circuito integrado MCP2003**

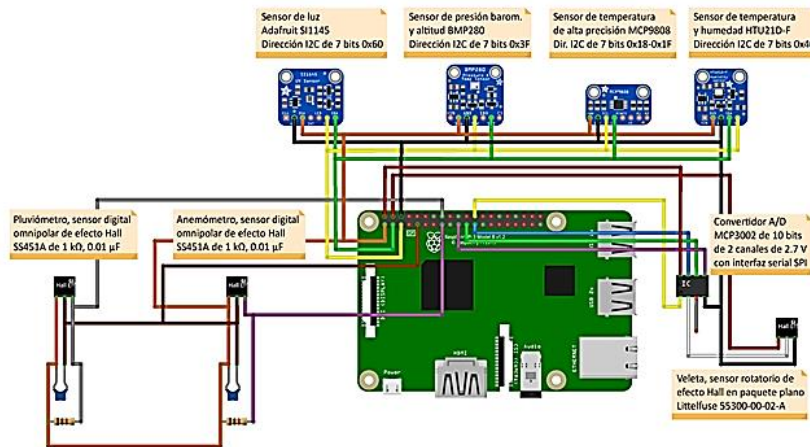


Fuente: elaboración propia, empleando Robot Structural 365.

4.3. Ensamblaje de sensores a *Raspberry Pi*

En este inciso se describe la unificación del hardware de la estación meteorológica, con el objetivo de establecer la comunicación entre el computador de tamaño reducido y sensores.

Figura 53. Esquema de conexión sensores, circuito integrado y *Raspberry Pi*

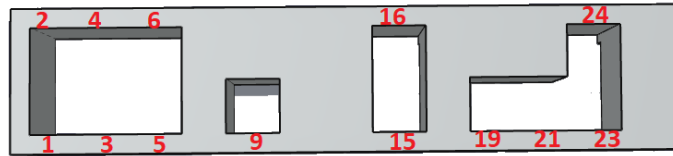


Fuente: elaboración propia, empleando Fritzing 0,8.3 B.

En la figura 54 se ejemplifica la conexión realizada entre los distintos sensores utilizados para la adquisición de datos y los diferentes componentes que permiten optimizar los GPIO disponibles de la *Raspberry Pi*. Por esta razón se da la necesidad de implementar un conector, impreso en 3D, el cual será acoplado a los GPIO de la *Raspberry Pi*.

En la figura 55 se muestra el diseño realizado y en la tabla XVIII se enlista la posición que corresponde a cada GPIO asociado.

Figura 54. **Diseño 3D de conector para GPIO Raspberry Pi**



Fuente: elaboración propia, empleando Microsoft PowerPoint 365.

Tabla XIX. **Posición de GPIO Raspberry Pi**

Posición	Función
1	3,3 V
2	5 V
3	I2C
4	5 V
5	I2C CLK
6	GND
9	GND
15	Datos velocidad de viento
16	Datos precipitación
19, 21, 23, 24	Señales SPI

Fuente: elaboración propia, empleando Microsoft Word 365.

Para el cableado interno de la estación meteorológica impresa en 3D, se utilizó cable de 3 hilos blindado estándar AWG 22, para los sensores SS451A, y cable de 4 hilos blindado estándar AWG 22, para los sensores BMP280, MC9808, HTU21D Y SI1145.

En la tabla XX se enlista la longitud utilizada para cada sensor.

Tabla XX. **Longitudes de cable utilizados para la estación meteorológica impresa en 3D**

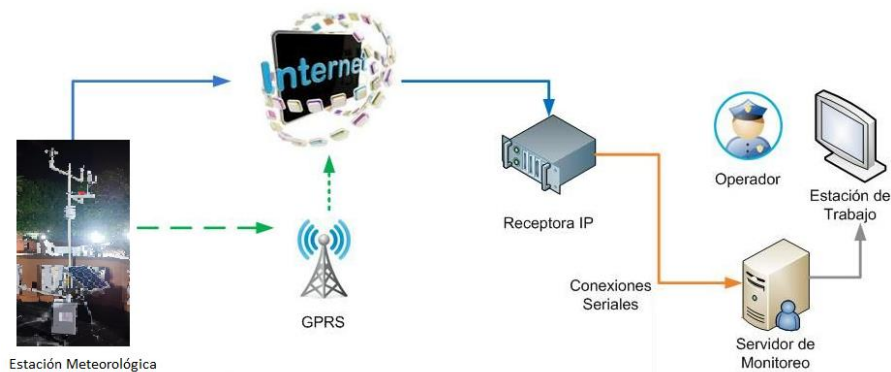
Sensor	Descripción	Unidad	Cantidad
SI1145	Cable de 4 hilos trenzado calibre AWG 22	cm	195
MCP9 808 (temperatura) BMP280 (presión) HTU21D (humedad)	Cable de 4 hilos trenzado, calibre AWG 22	cm	150
<i>Littelfuse</i> 55 300-00-02-A (anemómetro) SS451 (veleta)	Cable de 3 hilos trenzado, calibre AWG 22	cm	230
SI451A (pluviómetro)	Cable de 3 hilos trenzado, calibre AWG 22	cm	110

Fuente: elaboración propia, empleando Microsoft Excel 365.

4.4. Comunicación GPRS

Es conocido como servicio general de paquetes vía radio; que proporciona una cobertura inalámbrica y velocidades de transferencia de 56 kbps a 114 kbps.

Figura 55. **Diagrama de comunicación GPRS**



Fuente: elaboración propia, empleando Microsoft Visio 2016.

En la figura 56, se ejemplifica el proceso de comunicación de la estación meteorológica impresa en 3D, mediante el uso de GPRS. Como fue explicado en capítulos anteriores, GPRS es una extensión mejorada de FSM, que permite la mensajería instantánea, servicios de mensajes cortos (SMS) y multimedia (MMS). Permite una cobertura inalámbrica completa y velocidades de transferencia de entre 56 a 114 kbps.

En el capitulo 5, se detallará el funcionamiento y configuración utilizada para la comunicación, cabe resaltar que por motivos de seguridad informática no se detallará con un diagrama explícito la red interna de INSIVUMEH.

Se hace referencia que, en el diseño inicial, se estaba previsto mantener un microcontrolador Arduino NANO como esclavo y la *Raspberry Pi* como maestro para la transmisión de datos; esto se descartó con el objetivo de optimizar el tiempo de envío de datos hacia el servidor de INSIVUMEH.

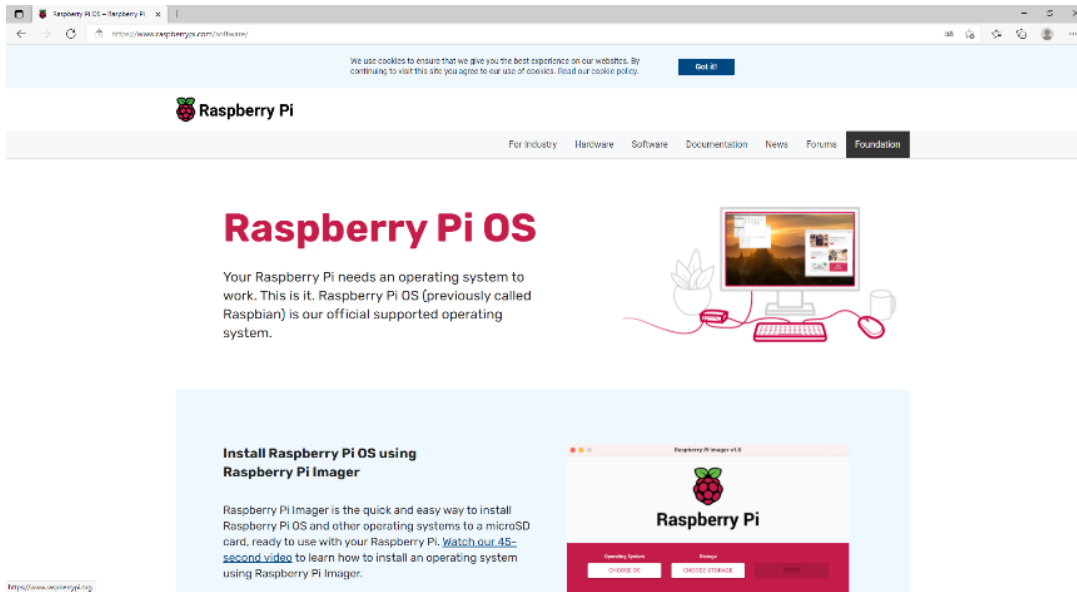
5. IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA APLICADO A UNA ESTACIÓN METEOROLÓGICA DE BAJO COSTO, MEDIANTE EL USO DE GPRS CON ACCESO A LA APN DE INSIVUMEH

En este capítulo se describe cada paso para poder realizar la implementación de *software* requerido para el correcto funcionamiento de una estación meteorológica impresa 3D, incluyendo el sistema de recolección, envío y recepción de datos, desde la estación hacia un equipo de cómputo en las oficinas centrales de INSIVUMEH.

5.1. Sistema operativo *Raspberry Pi*

Para descargar la versión más actualizada de Raspbian, es necesario ingresar al sitio web oficial de *Raspberry Pi* (<https://www.raspberrypi.com/software/>). En la figura 57 se muestra el inicio de dicha página web.

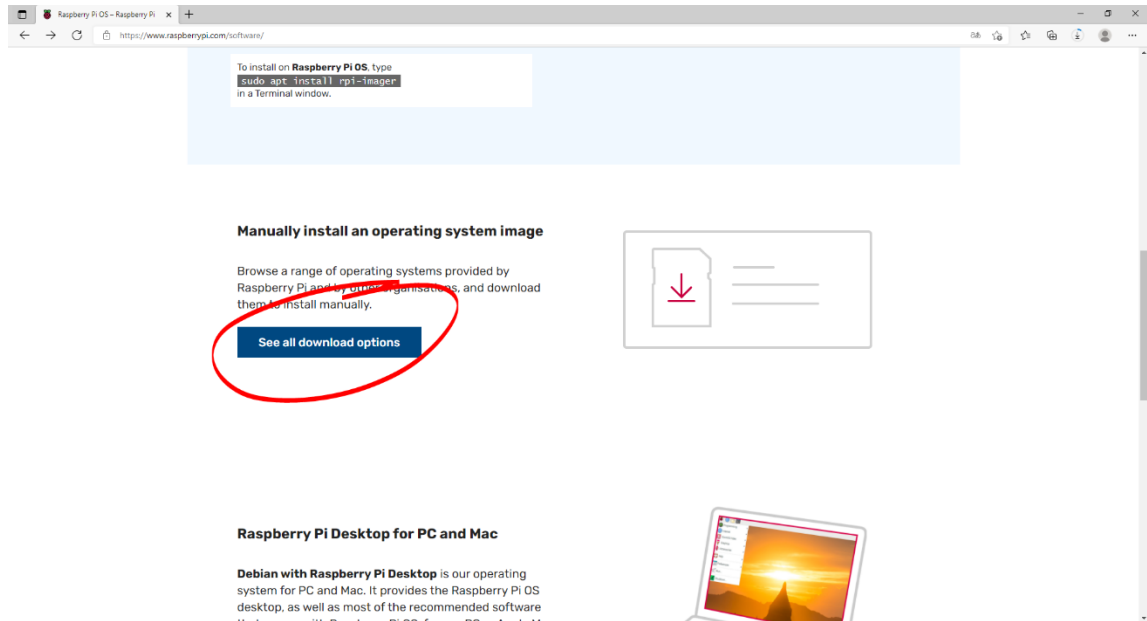
Figura 56. Sitio web oficial *Raspberry Pi* para descarga de software



Fuente: elaboración propia, empleando captura de pantalla.

Seguidamente nos desplazamos a la parte inferior de la página web, y descargamos la imagen según nuestro sistema operativo para realizar el montaje de la imagen en nuestra microSD, ya descrita en el capítulo dos. En esta ocasión seleccionamos la opción que nos permite ver todas las opciones de descarga. En la figura 58 se resalta la opción a seleccionar.

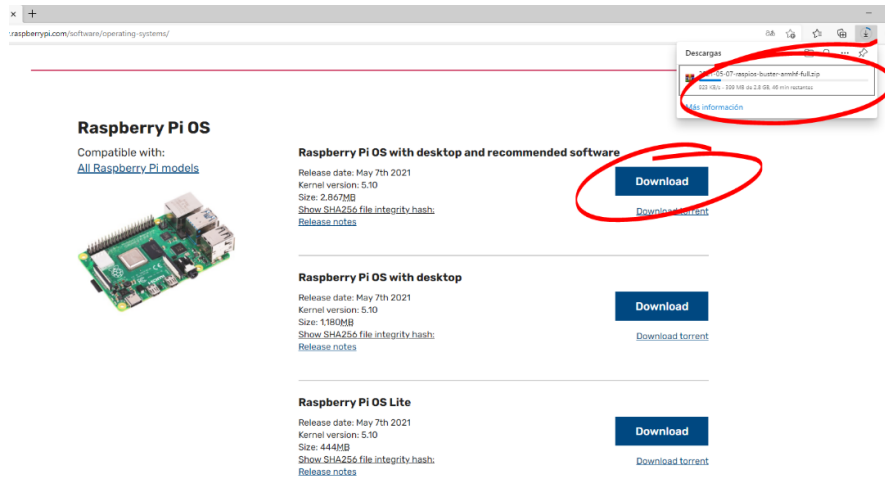
Figura 57. Botón para ver todas las opciones de descarga



Fuente: elaboración propia, empleando captura de pantalla.

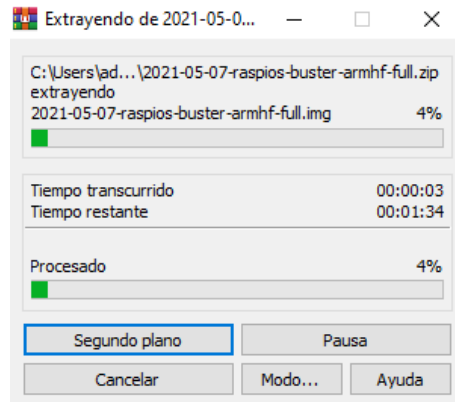
Seguidamente la opción seleccionada redirección a la sección de *Raspberry Pi OS*. Seleccionamos la opción de *Raspberry Pi OS* con desktop y *software* recomendado por los desarrolladores del *software*, seguidamente iniciará la descarga de un archivo .zip, dicha acción se puede apreciar en la figura 59, el cual contiene el archivo .img para ser agregada a nuestra micro SD; para ello descomprimos la carpeta.zip.

Figura 58. Descarga imagen de *Raspberry Pi OS*



Fuente: elaboración propia, empleando captura de pantalla.

Figura 59. Descomprimir archivo.zip



Fuente: elaboración propia, empleando captura de pantalla.

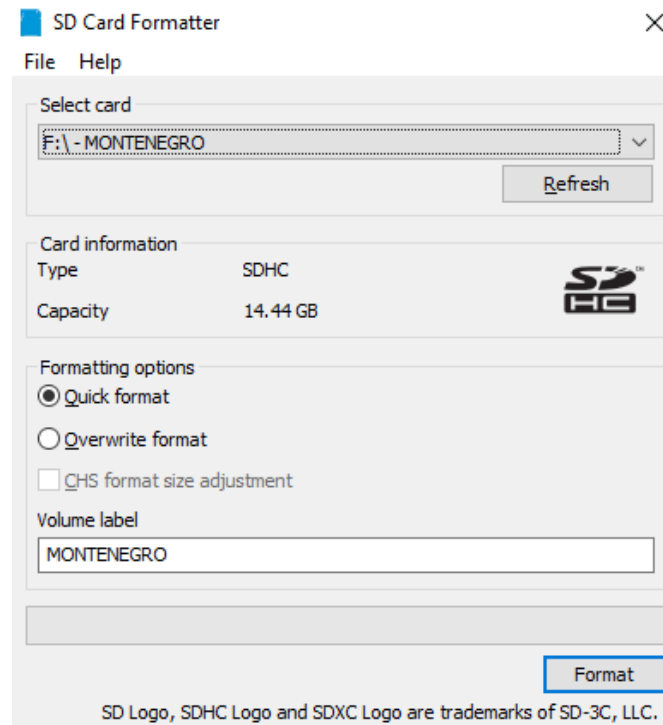
Para el siguiente paso necesitaremos de una microSD, para este trabajo se utilizó una SD de 32 GB de capacidad de almacenamiento, y un adaptador a SD; y la aplicación *SD Card Formatter*.

Figura 60. **Micro SD y adaptador**



Fuente: elaboración propia. INSIVUMEH.

Figura 61. **Aplicación SD Card Formatter**

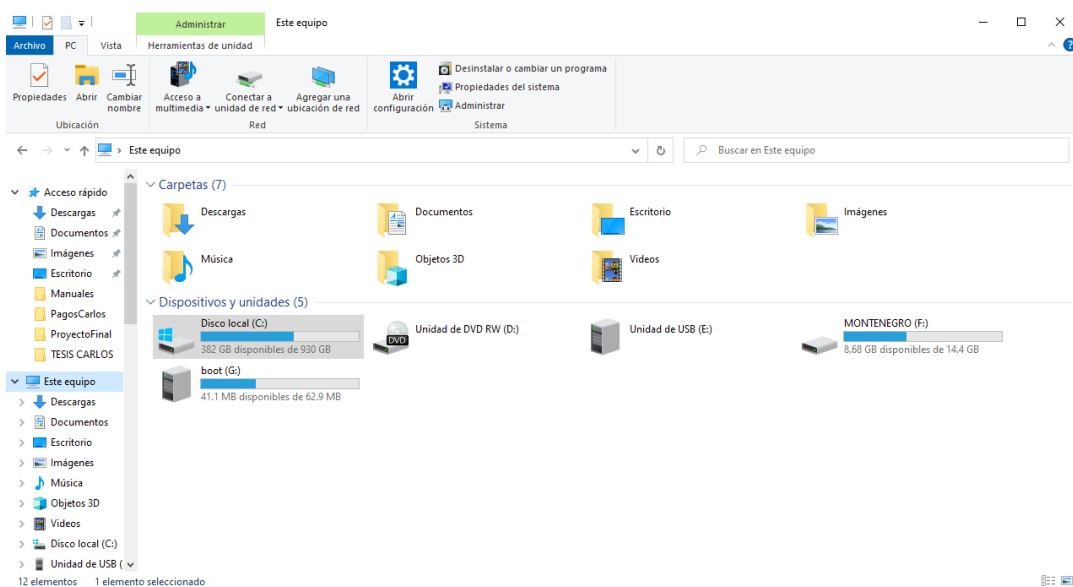


Fuente: elaboración propia, empleando captura de pantalla.

Ingresamos nuestra microSD dentro de nuestro adaptador y lo introducimos en el puerto de nuestro ordenador para proceder a formatearlo. Verificamos los dispositivos que ha reconocido nuestro ordenador, para evitar inconvenientes al momento de formatear nuestra microSD. En la aplicación *SD Card Formatter* seleccionamos el dispositivo deseado a formatear y verificamos en la sección de opciones de formateo que se encuentre en un formato rápido (*Quick format*).

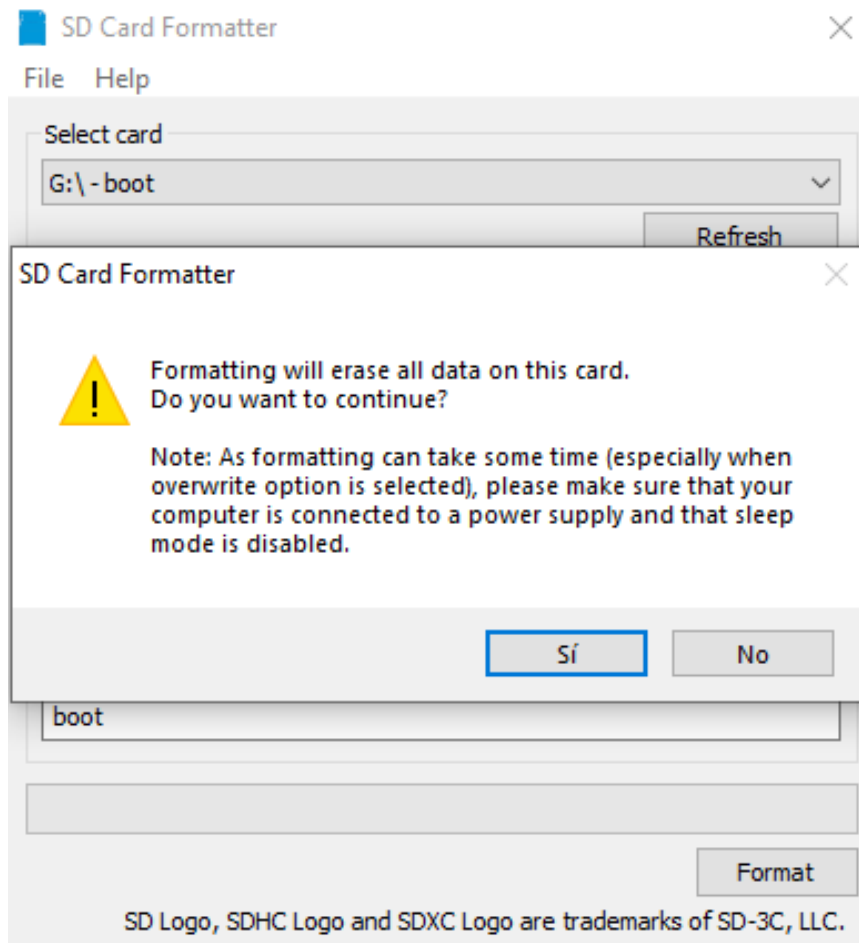
La aplicación *SD Card Formatter* permite nombrar, por medio de una etiqueta, el volumen de nuestra microSD. Verificando la información, se procede a presionar el botón *Format*, donde se mostrará un mensaje de advertencia, que al momento de realizar el formateo perderemos la información del equipo de almacenamiento y se procede aceptando para continuar. Se espera unos segundos en lo que se ejecuta la acción. Ver figura 64.

Figura 62. **Hardware reconocido por el ordenador**



Fuente: elaboración propia, empleando captura de pantalla.

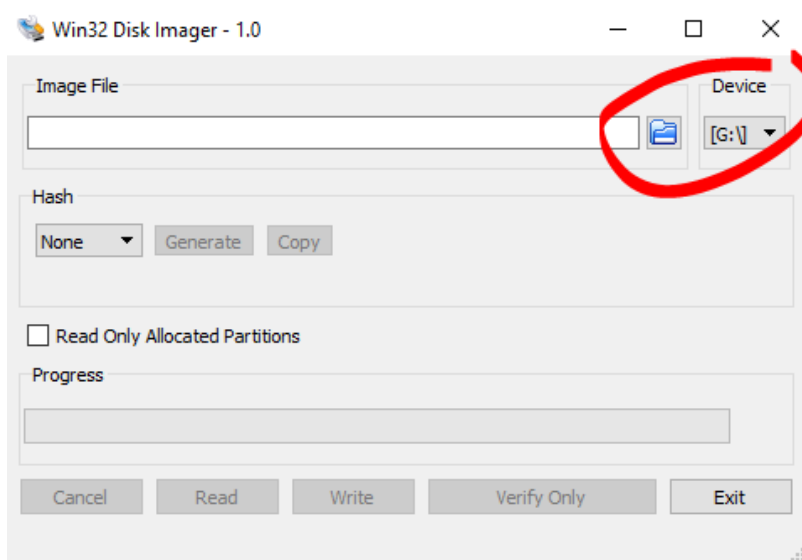
Figura 63. **Mensaje de advertencia al formatear microSD**



Fuente: elaboración propia, empleando captura de pantalla.

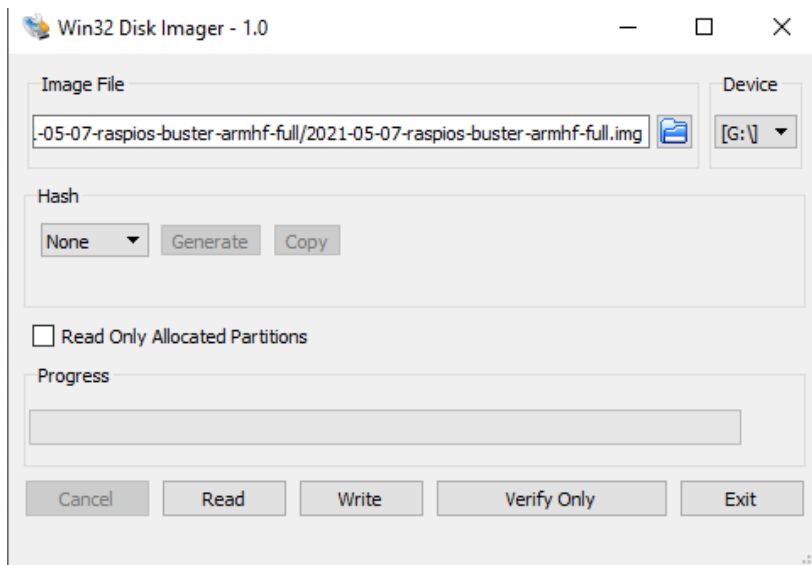
Para el siguiente paso, necesitamos la aplicación Win32 Disk Imager, esta aplicación nos permitirá agregar la imagen de *Raspberry Pi OS* a nuestra microSD formateada. Es necesario verificar a que dispositivo agregaremos el archivo .img, en la parte superior derecha de la interfaz gráfica de la aplicación Win32 Disk Imager y se procede a buscar el archivo .img presionando en el folder de color azul, ver figuras 65 y 66.

Figura 64. Win32 Disk Imager



Fuente: elaboración propia, empleando captura de pantalla.

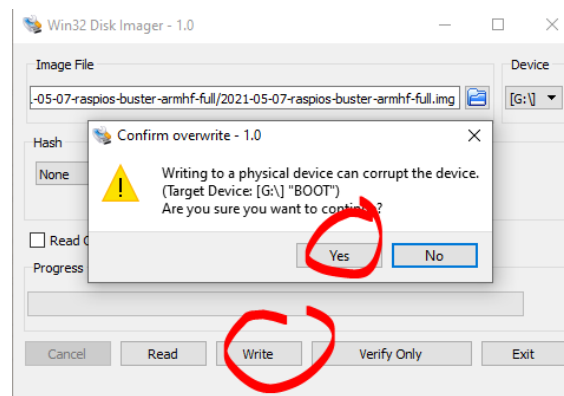
Figura 65. Agregar archivo .img a Win32 Disk Imager



Fuente: elaboración propia, empleando captura de pantalla.

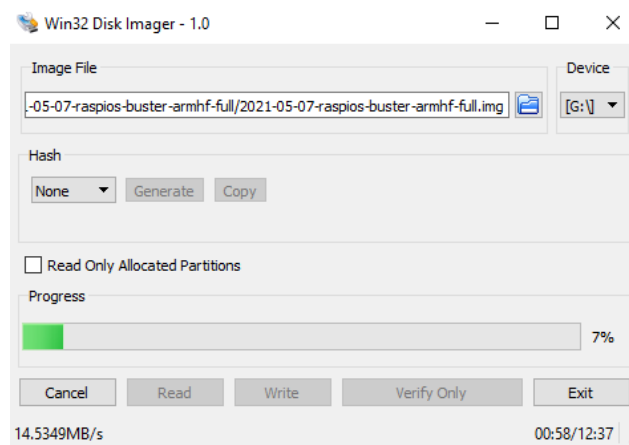
Presionamos el botón de *write*, el cual permite escribir el archivo seleccionado sobre nuestra microSD, seguidamente mostrará un mensaje de advertencia y aceptamos para continuar. Iniciaré el proceso de agregar la imagen en la micoSD. Ver figuras 67 y 68.

Figura 66. **Mensaje de advertencia al escribir archivo .img en microSD**



Fuente: elaboración propia, empleando captura de pantalla.

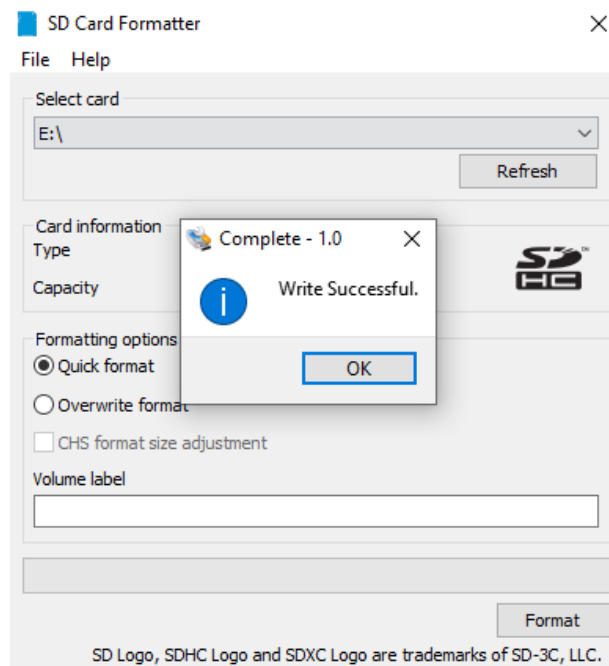
Figura 67. **Proceso de escritura archivo .img en micro SD**



Fuente: elaboración propia, empleando captura de pantalla.

Al finalizar con el proceso de grabado de imagen, aparecerá un mensaje con la leyenda *Write Successful*. Ver figura 69.

Figura 68. **Finalización de grabado archivo .img en microSD**



Fuente: elaboración propia, empleando captura de pantalla.

5.2. Configuración *Raspberry Pi*

Al finalizar el grabado del archivo .img en nuestra microSD, procedemos a ingresarla en el *slot* de nuestra *Raspberry Pi*.

Para alimentar nuestra RPI utilizaremos un adaptador micro USC de 5V, con corriente de salida de 1,2 A (aunque con 1 000mA es suficiente para su alimentación al momento de realizar las configuraciones iniciales de nuestro sistema operativo).

Para interactuar con la interfaz gráfica, como primer paso, conectaremos en los puertos USB de nuestra *Raspberry Pi* un teclado y un *mouse*, al igual que un cable HDMI al puerto correspondiente, conectado a una pantalla. Ver figura 70.

Figura 69. **Interacción con la interfaz gráfica de *Raspberry Pi***



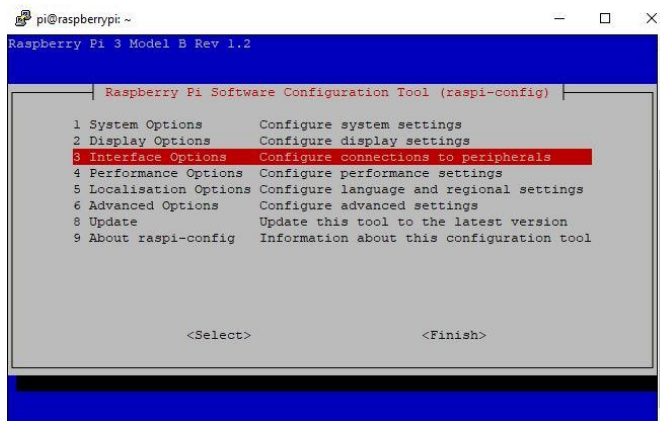
Fuente: elaboración propia. INSIVUMEH.

Al iniciar el sistema operativo, aparece un mensaje de bienvenida y realizamos los siguientes pasos:

- Presionamos siguiente para avanzar.
- Realizamos la configuración de nuestro país, idioma y zona horaria.
- Presionamos siguiente y esperamos a que se realice el cambio correspondiente.
- Podemos agregar una contraseña nueva, ya que las credenciales por defecto se enlistan a continuación.

- Usuario: pi
- Contraseña: Raspberry
- Se realiza la configuración de nuestra pantalla.
- Seleccionamos alguna red Wifi disponible, o simplemente podemos conectaros por LAN a un *Router* o *switch*.
- Habilitamos el acceso por SSH, ingresando el comando “sudo raspi-config”.
- Se desplegará una nueva ventana, en el cual se puede desplazar utilizando las flechas del teclado, e ingresa al inciso 3 (*Interface Options*) presionando la tecla *Enter*. Ver figura 70.
- Seguido seleccionamos la opción de SSH, presionamos *enter* y nos mostrará una pantalla indicando que ha sido habilitado dicha interfaz.
- Seleccionamos la opción finalizar.

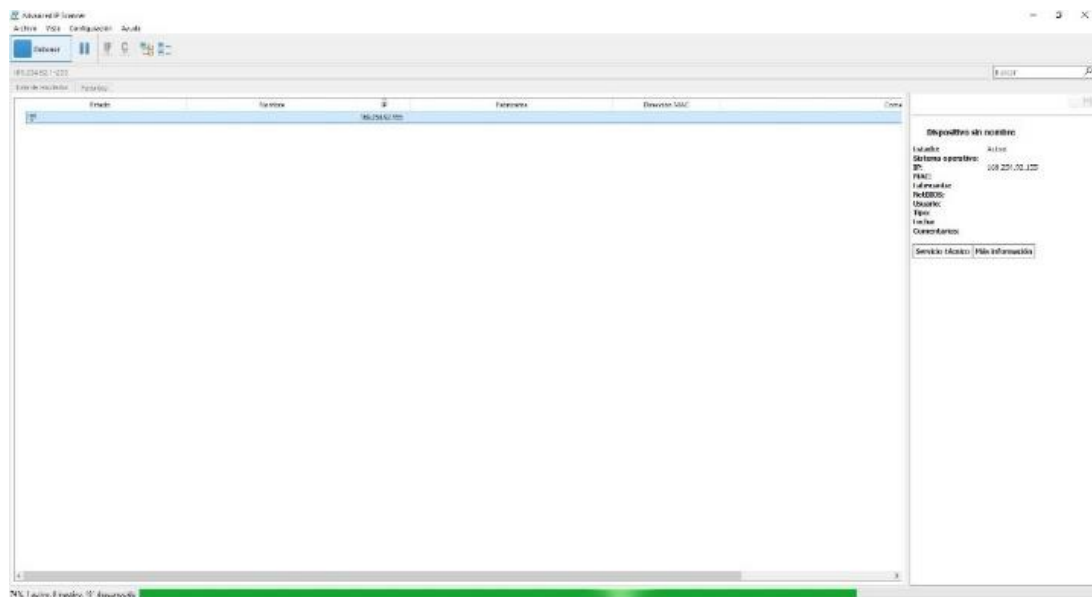
Figura 70. **Interface Options de Raspberry Pi**



Fuente: elaboración propia, empleando captura de pantalla.

Finalizado con el proceso de configuración, podemos realizar nuestra conexión a la *Raspberry Pi* mediante el protocolo SSH. Para ello utilizaremos el programa Putty (Ver figura 73) o podemos ingresar por consola de Windows (ver figura 72). Para conocer la IP de nuestro dispositivo, se realiza un escaneo de la red, en este caso utilizaremos el programa *Advanced IP Scanner* (ver figura 72).

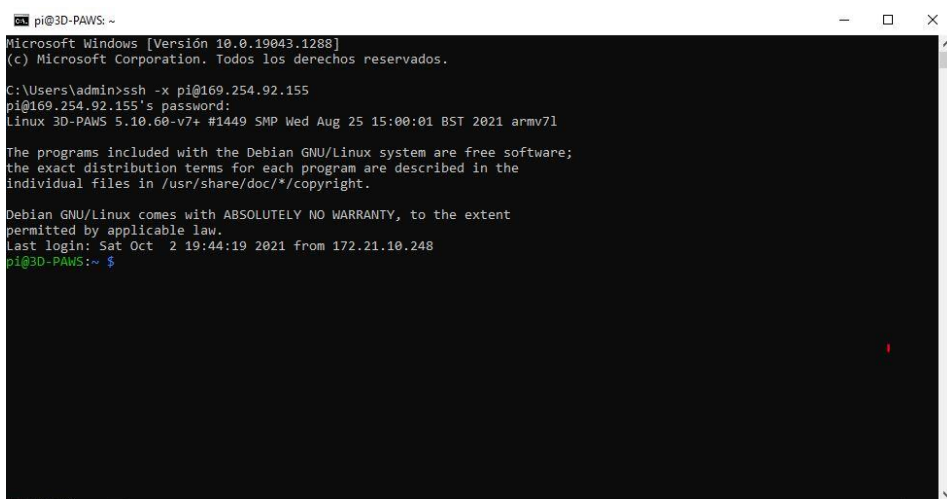
Figura 71. **Utilización *Advanced IP Scanner***



Fuente: elaboración propia, empleando *Advance IP Scanner* y captura de pantalla.

Al realizar el escaneo de nuestra red y conocer la IP de nuestro dispositivo realizamos nuestra conexión SSH.

Figura 72. Comunicación SSH desde cmd



```
pi@3D-PAWS: ~
Microsoft Windows [Versión 10.0.19043.1288]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Admin>ssh -x pi@169.254.92.155
pi@169.254.92.155's password:
Linux 3D-PAWS 5.10.60-v7+ #1449 SMP Wed Aug 25 15:00:01 BST 2021 armv7l

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: Sat Oct  2 19:44:19 2021 from 172.21.10.248
pi@3D-PAWS:~$
```

Fuente: elaboración propia, empleando captura de pantalla.

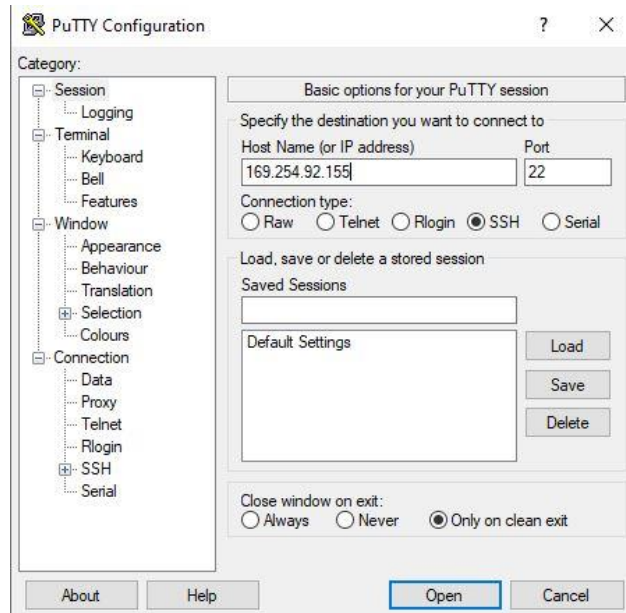
El comando utilizado para ingresar por SSH a nuestro dispositivo desde la terminal de Windows es:

```
ssh -X pi@169,254.92,155
```

Donde ssh es el protocolo utilizado, -x habilita la interfaz gráfica (no es necesario colocarlo), pi es el usuario de nuestro dispositivo y la IP seguida del símbolo @ es la IP de nuestro dispositivo.

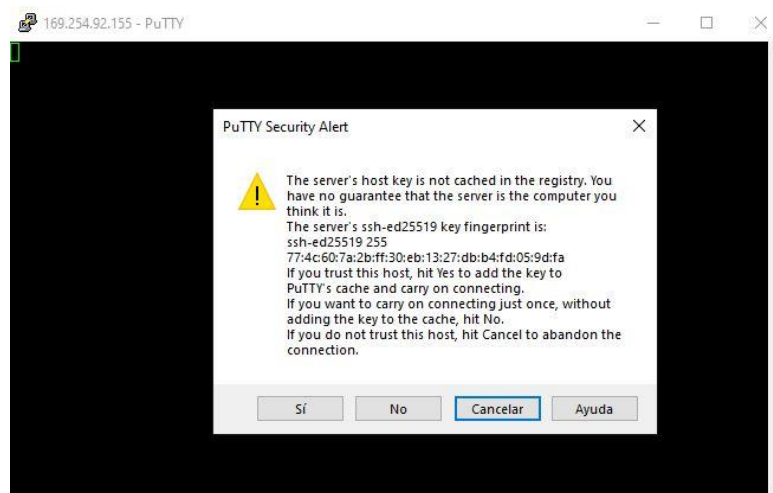
Si es complicado realizar dicha conexión, se puede apoyar con el programa Putty, el cual permite realizar conexiones por diferentes protocolos.

Figura 73. Programa Putty



Fuente: elaboración propia, empleando captura de pantalla.

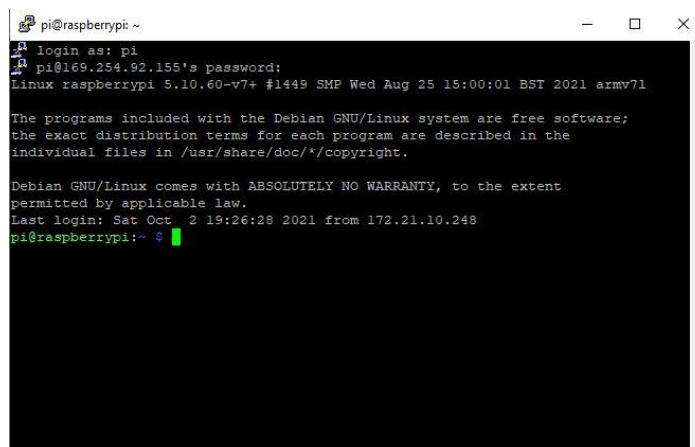
Figura 74. Alerta Putty



Fuente: elaboración propia, empleando captura de pantalla.

Al momento de ingresar por ssh a la *Raspberry Pi*, Putty mostrará un mensaje de advertencia, presionamos el botón si, y nos desplegará la terminal de nuestro dispositivo, en el cual debemos ingresar usuario y contraseña, establecida por nosotros en los pasos de configuración inicial.

Figura 75. Ingreso por ssh utilizando Putty



```
pi@raspberrypi ~
login as: pi
pi@169.254.92.155's password:
Linux raspberrypi 5.10.60-v7+ #1449 SMP Wed Aug 25 15:00:01 BST 2021 armv7l

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: Sat Oct 2 19:26:28 2021 from 172.21.10.248
pi@raspberrypi:~$
```

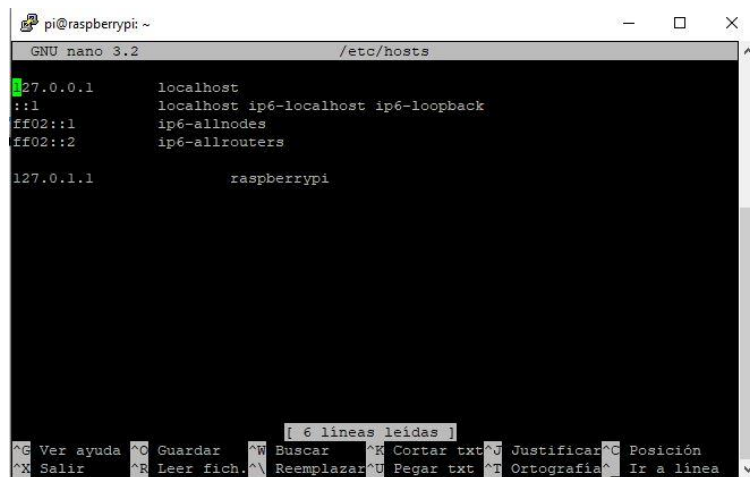
Fuente: elaboración propia, empleando captura de pantalla.

Continuando con la configuración inicial de la *Raspberry Pi*, podemos realizar el cambio de nombre de nuestro dispositivo, para este trabajo asignaremos el nombre 3D-PAWS, esto se realiza ingresando como super usuario, y cambiando el nombre en los siguientes ficheros.

- Cambio a super usuario: ingresar en la terminal el comando `sudo su`.
- Editar el archivo `hosts`, ubicado en `/etc/hosts`, utilizando: `sudo nano /etc/hosts`.
 - En esta opción, cambiamos el nombre de *Raspberry pi* por el deseado, ver figuras 76 y 77.

- Editar el archivo *hostname*, ubicado en */etc/hostname*, utilizando `sudo nano /etc/hostname`.
 - En esta opción, cambiamos el nombre *Raspberry pi* por el nombre deseado, ver figuras 78 y 79.
- Reiniciamos el dispositivo con el comando: `sudo reboot`, o se puede realizar un apagado del equipo con el comando: `shutdown -r now`.

Figura 76. Fichero */etc/hosts*



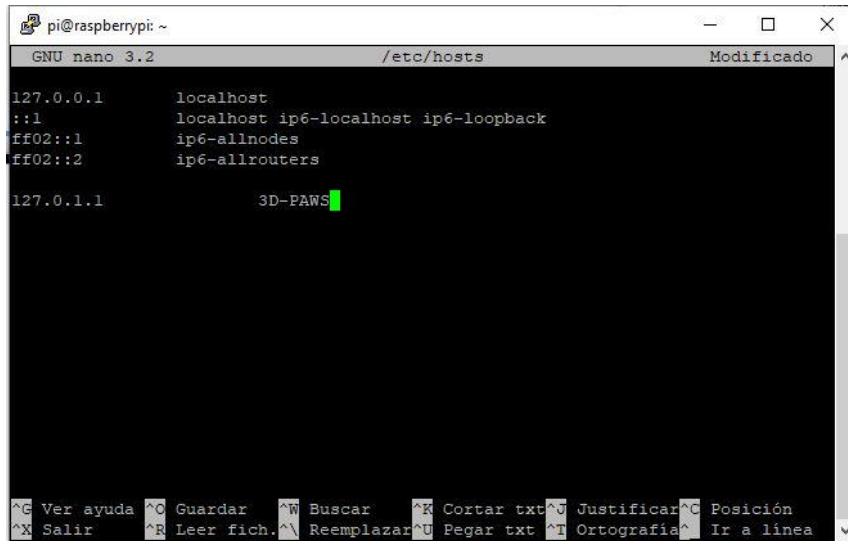
```
pi@raspberrypi: ~
GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

127.0.1.1    raspberry

[ 6 líneas leídas ]
^G Ver ayuda  ^C Guardar  ^W Buscar   ^K Cortar txt ^J Justificar ^C Posición
^X Salir     ^R Leer fich. ^_ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Fuente: elaboración propia, empleando captura de pantalla.

Figura 77. Cambio de nombre en fichero *hosts*



The screenshot shows a terminal window on a Raspberry Pi. The nano editor is open to the file `/etc/hosts`. The file content is as follows:

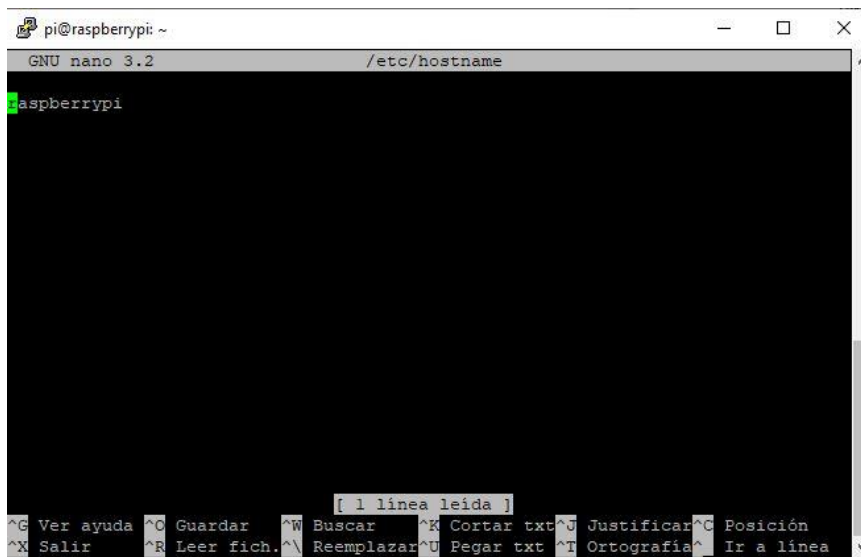
```
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

127.0.1.1   3D-PAWS
```

The cursor is positioned at the end of the line `127.0.1.1 3D-PAWS`. The nano editor's status bar at the bottom shows various keyboard shortcuts.

Fuente: elaboración propia, empleando captura de pantalla.

Figura 78. Fichero */etc/hostname*



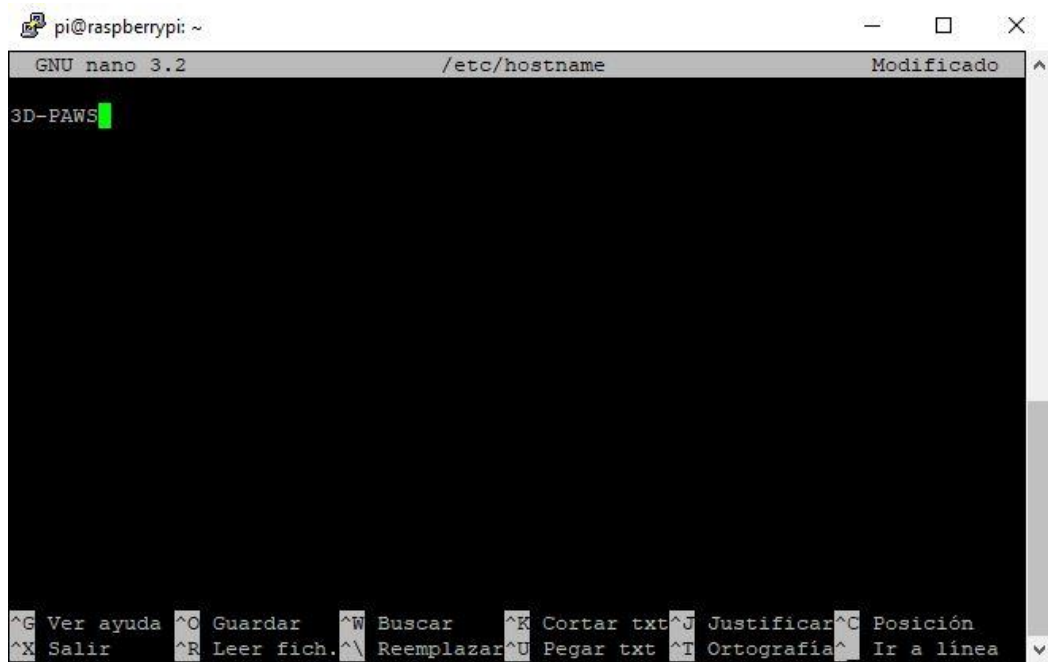
The screenshot shows a terminal window on a Raspberry Pi. The nano editor is open to the file `/etc/hostname`. The file content is:

```
raspberrypi
```

The cursor is at the end of the line. A status bar above the nano editor indicates `[1 línea leída]`. The nano editor's status bar at the bottom shows various keyboard shortcuts.

Fuente: elaboración propia, empleando captura de pantalla.

Figura 79. Cambio de nombre en fichero *hostname*



```
pi@raspberrypi: ~  
GNU nano 3.2 /etc/hostname Modificado  
3D-PAWS  
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^I Ir a línea
```

Fuente: elaboración propia, empleando captura de pantalla.

Es necesario conocer que, al momento de realizar las modificaciones, para salir de los ficheros ingresados, se presiona el comando de teclado Ctrl + X y pulsamos la letra S, seguido de *Enter*.

Luego de haber realizado la configuración inicial de *Raspberry Pi*, realizamos la actualización de paquetes instalados en el dispositivo, que contiene los programas, bibliotecas y controladores básicos para continuar. Realizamos los siguientes pasos:

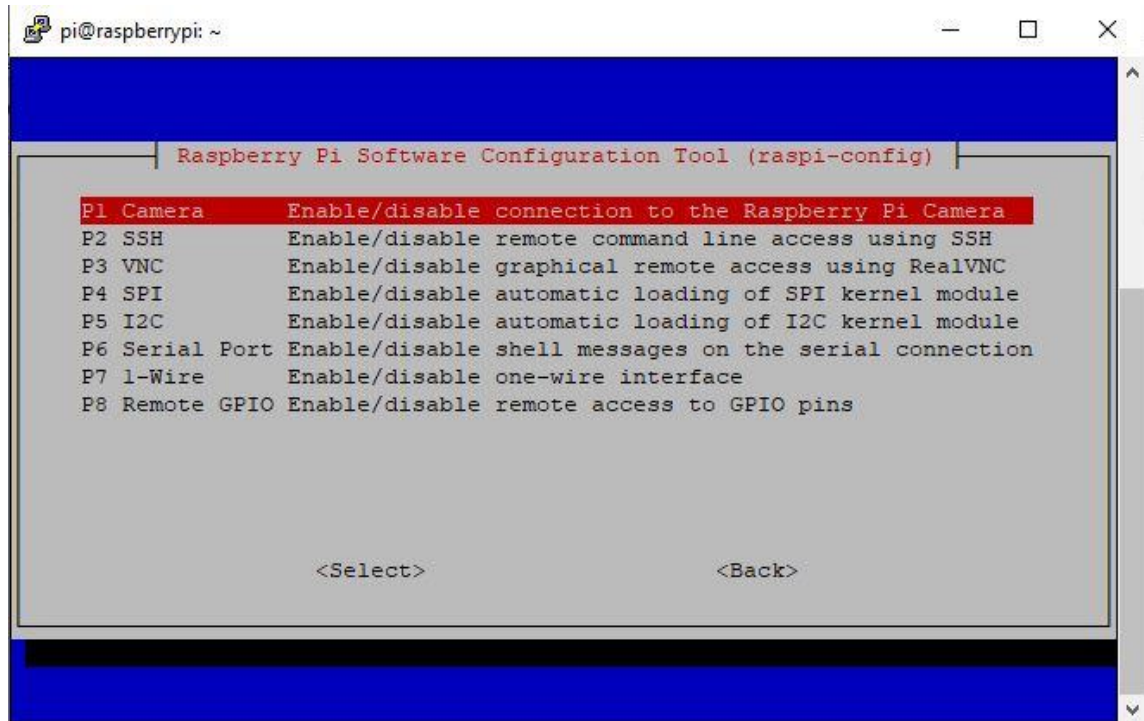
- Abrimos la terminal de *Raspberry Pi* con el comando Ctrl + Alt + t.
- Ingresa el comando “*sudo apt-get update*” y presionamos *Enter*.

- El minicomputador iniciara a descargar la nueva lista de paquetes para verificar y validar los paquetes disponibles en su versión más reciente.
- Al finalizar de ejecutarse el comando anterior, se procede a ingresar el comando “*sudo apt-get upgrade*” y presionamos *Enter*.
 - Antes de instalar los nuevos paquetes con el comando instalado previamente, se desplegará un mensaje sobre el espacio de almacenamiento requerido para ser ejecutado. Si no se cuenta con espacio suficiente, se debe cancelar la actualización ingresando “N” y proporcionar espacio de almacenamiento adicional.
 - Si se posee espacio suficiente, se realiza la actualización presionando la tecla “S”.
- Al finalizar la actualización, se reinicia la *Raspberry Pi* con el comando “*sudo reboot*”.

Con los paquetes actualizados, se procede a habilitar las interfaces necesarias para la implementación de los sensores a utilizar. Se escribe en la terminal de comandos “*sudo raspi-config*”. Ver figura 71.

- Seleccionamos la opción tres (*Interface Options*).
- Se habilita SPI.
- Se habilita la interfaz I2C.
- Se habilita la interfaz Serial Port.
- Se habilita la interfaz VNC (opcional) que nos permitirá tener un acceso remoto con interfaz gráfica del sistema operativo de *Raspberry Pi*.

Figura 80. Interfaces de *Raspberry Pi*



Fuente: elaboración propia, empleando captura de pantalla.

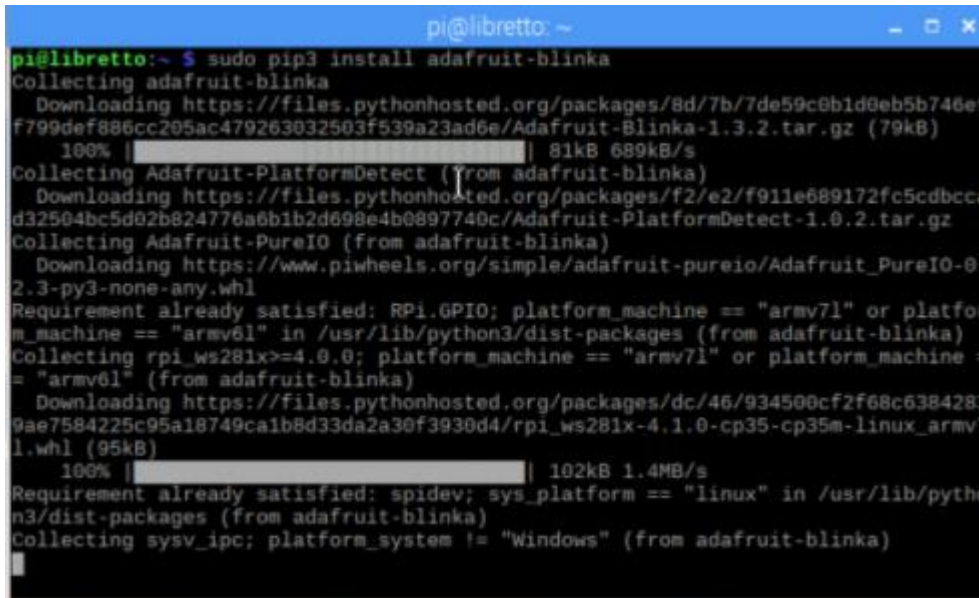
5.2.1. Código Pi de Adafruit

Para facilitar el proceso de conexión de los sensores, Adafruit ha creado una colección de código extensa y útil; esto incluye bibliotecas *Circuito Python* simples para una gran cantidad de módulos, incluidas pantallas, sensores, actuadores y más.

Para obtener este código, se debe utilizar un *software* llamado “git”.

Ejecutaremos el siguiente comando para instalar las librerías Adafruit_blinka CircuitPython. “pip3 intall adafruit flash”.

Figura 81. **Pip3 install adafruit flash**



```
pi@libretto: ~  
pi@libretto:~$ sudo pip3 install adafruit-blinka  
Collecting adafruit-blinka  
  Downloading https://files.pythonhosted.org/packages/8d/7b/7de59c8b1d0eb5b746ef  
f799def886cc295ac479263032503f539a23ad6e/Adafruit-Blinka-1.3.2.tar.gz (79kB)  
  100% |#####| 81kB 689kB/s  
Collecting Adafruit-PlatformDetect (from adafruit-blinka)  
  Downloading https://files.pythonhosted.org/packages/f2/e2/f911e689172fc5cdbcca  
d32504bc5d02b824770a6b1b2d698e4b0897740c/Adafruit-PlatformDetect-1.0.2.tar.gz  
Collecting Adafruit-PureIO (from adafruit-blinka)  
  Downloading https://www.piwheels.org/simple/adafruit-pureio/Adafruit_PureIO-0.  
2.3-py3-none-any.whl  
Requirement already satisfied: RPi.GPIO; platform_machine == "armv7l" or platfor  
m_machine == "armv6l" in /usr/lib/python3/dist-packages (from adafruit-blinka)  
Collecting rpi_ws281x>=4.0.0; platform_machine == "armv7l" or platform_machine =  
= "armv6l" (from adafruit-blinka)  
  Downloading https://files.pythonhosted.org/packages/dc/46/934500cf2f68c6384283  
9ae7584225c95a18749ca1b8d33da2a30f3930d4/rpi_ws281x-4.1.0-cp35-cp35m-linux_armv7  
l.whl (95kB)  
  100% |#####| 102kB 1.4MB/s  
Requirement already satisfied: spidev; sys_platform == "linux" in /usr/lib/pytho  
n3/dist-packages (from adafruit-blinka)  
Collecting sysv_ipc; platform_system != "Windows" (from adafruit-blinka)
```

Fuente: Learn Adafruit. *Adafruit pi code*. <https://learn.adafruit.com/adafruit-Raspberry-pi-lesson-4-gpio-setup/adafruit-pi-code>. Consulta: 16 de octubre de 2021.

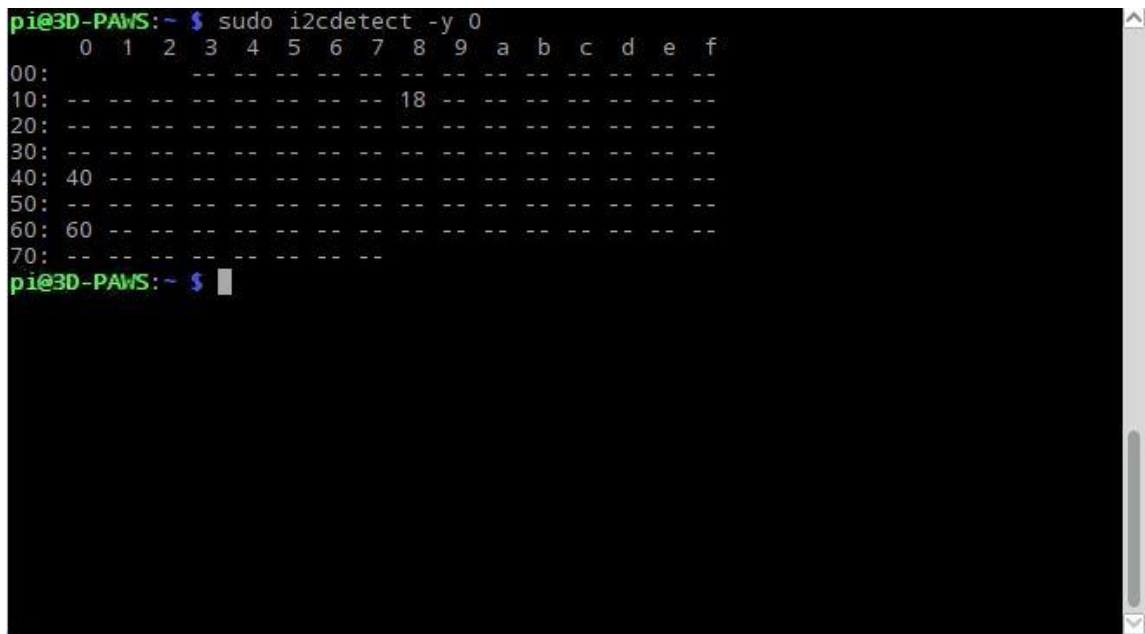
5.2.2. Prueba I2C

I2C es un estándar de uso muy común diseñado para permitir que un chip se comunique con otro. Entonces, dado que la *Raspberry Pi* puede comunicarse mediante I2C, podemos conectarla a una variedad de módulos compatibles con I2C.

El bus I2C permite conectar varios dispositivos a la *Raspberry Pi*, cada uno con una dirección única, que a menudo se puede configurar cambiando la configuración del puente en el módulo. Es muy útil poder ver qué dispositivos están conectados.

Para comprobar los dispositivos conectados utilizamos el comando “sudo i2cdetect -y 0”. Ver figura 83.

Figura 82. **Comando sudo i2cdetect -y 0**



```
pi@3D-PAWS:~ $ sudo i2cdetect -y 0
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- 18 -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@3D-PAWS:~ $
```

Fuente: elaboración propia, empleando captura de pantalla.

5.2.3. **Prueba SPI**

El bus SPI es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos, el cual es un estándar para controlar dispositivos digitales que acepte un flujo de bits en serie regulado por un reloj.

Para comprobar los dispositivos que están en uso, se utiliza el comando “ls -l /dev/spidev*”. Ver figura 84.

Figura 83. Comando `ls -l /dev/spidev*`

```
pi@3D-PAWS:~$ ls -l /dev/spidev*
crw-rw---- 1 root spi 153, 0 Oct 10 02:17 /dev/spidev0.0
crw-rw---- 1 root spi 153, 1 Oct 10 02:17 /dev/spidev0.1
pi@3D-PAWS:~$ █
```

Fuente: elaboración propia, empleando captura de pantalla.

5.2.4. *Instalación de Python*

El lenguaje de programación *Python* en realidad inició como un lenguaje de *scripting* para Linux. Los programas d *Python* son similares a los *scripts* de Shell en que los archivos contienen una seria de comandos que la computadora ejecuta de arriba abajo.

Al igual que los *scripts* de Shell, *Python* puede automatizar tareas como cambiar el nombre y mover archivos. Se puede utilizar como una línea de comandos con IDLE, la función REPL (leer, eval, imprimir, bucle) de *Python*.

Python posee bibliotecas para distintas finalidades, como programación de juegos, gráficos de representación, interfaces GUI marcos web y computación científica.

Regularmente *Python 2* y *Python 3* se encuentran preinstalados en los sistemas operativos Raspbian, pero para instalarlo o actualizarlo (según sea el caso) simplemente se ejecutan los comandos en la terminal de *Raspberry Pi*:

- Instalar o actualizar *Python 2*: “*sudo apt-get install Python*”.
- Instalar o actualizar *Python 3*: “*sudo apt-get install Python3*”.

5.3. Programación de sensores

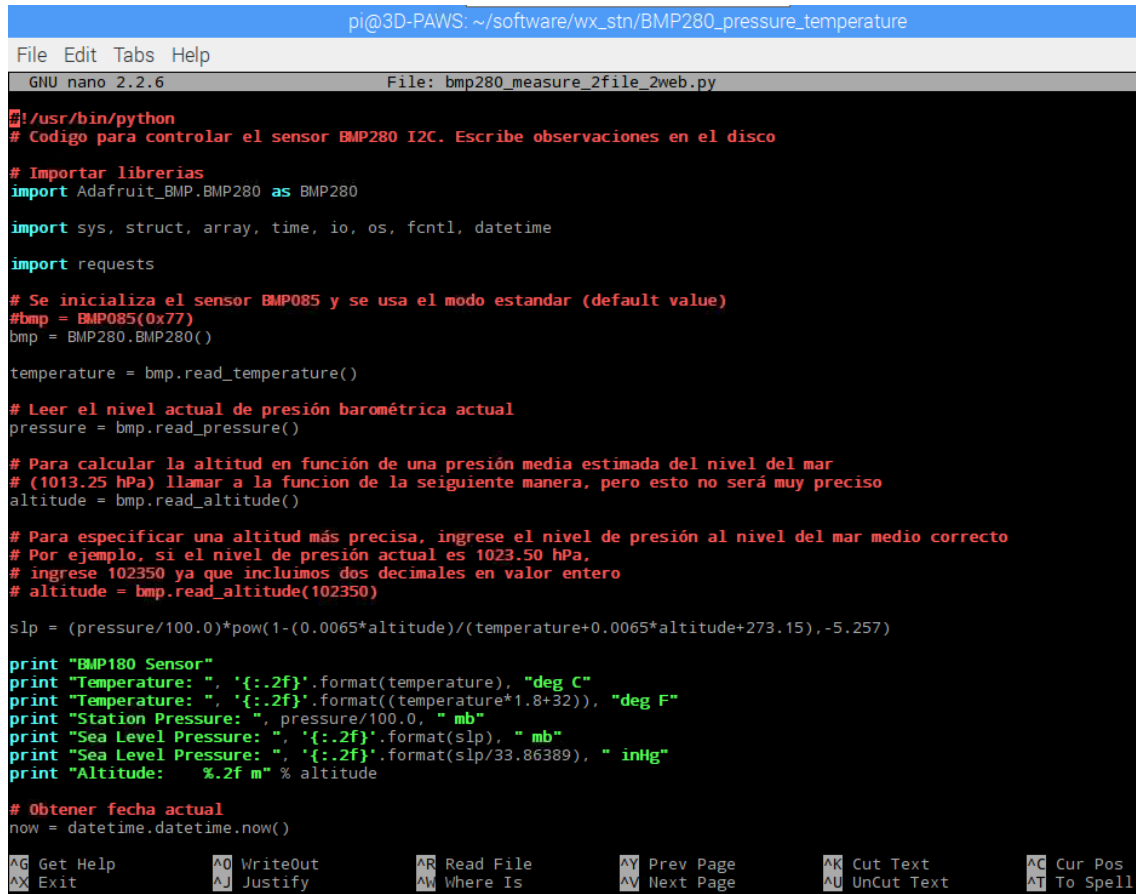
Raspberry Pi es capaz de medir una cantidad de sensores digitales y analógicos con horarios definidos y capacidad de almacenamiento o transmisión de datos a la nube.

En esta sección se mostrará el desarrollo de cada sensor incorporado a la *Raspberry Pi* para su correcto funcionamiento, lectura de datos y almacenamiento en el espacio disponible de la memoria microSD.

5.3.1. Sensor de presión BMP280

En la figura 85, se observa la primera sección del código implementado, en el cual se crea una línea de *Shebang*, la cual define dónde se encuentra el intérprete, para este sensor utilizamos *Python2*, el cual establece el indicador de ejecución de un *script*. También se importan las librerías a utilizar, la inicialización del sensor, lectura de presión actual, impresión en pantalla al momento de ejecutar el programa de forma individual y la obtención de la fecha actual, para posteriormente crear un archivo diario para la base de datos.

Figura 84. Código sensor BMP280 parte 1



```
pi@3D-PAWS: ~/software/wx_stn/BMP280_pressure_temperature
File Edit Tabs Help
GNU nano 2.2.6 File: bmp280_measure_2file_2web.py
#!/usr/bin/python
#Codigo para controlar el sensor BMP280 I2C. Escribe observaciones en el disco
# Importar librerias
import Adafruit_BMP.BMP280 as BMP280
import sys, struct, array, time, io, os, fcntl, datetime
import requests
# Se inicializa el sensor BMP085 y se usa el modo estandar (default value)
#bmp = BMP085(0x77)
bmp = BMP280.BMP280()
temperature = bmp.read_temperature()
# Leer el nivel actual de presión barométrica actual
pressure = bmp.read_pressure()
# Para calcular la altitud en función de una presión media estimada del nivel del mar
# (1013.25 hPa) llamar a la función de la siguiente manera, pero esto no será muy preciso
altitude = bmp.read_altitude()
# Para especificar una altitud más precisa, ingrese el nivel de presión al nivel del mar medio correcto
# Por ejemplo, si el nivel de presión actual es 1023.50 hPa,
# ingrese 102350 ya que incluimos dos decimales en valor entero
# altitude = bmp.read_altitude(102350)
slp = (pressure/100.0)*pow(1-(0.0065*altitude)/(temperature+0.0065*altitude+273.15),-5.257)
print "BMP180 Sensor"
print "Temperature: ", '{:.2f}'.format(temperature), "deg C"
print "Temperature: ", '{:.2f}'.format((temperature*1.8+32)), "deg F"
print "Station Pressure: ", pressure/100.0, " mb"
print "Sea Level Pressure: ", '{:.2f}'.format(slp), " mb"
print "Sea Level Pressure: ", '{:.2f}'.format(slp/33.86389), " inHg"
print "Altitude: %.2f m" % altitude
# Obtener fecha actual
now = datetime.datetime.now()
AG Get Help      AO WriteOut     AR Read File    AY Prev Page    AK Cut Text     AC Cur Pos
AX Exit          AJ Justify      AW Where Is     AN Next Page    AU UnCut Text   AT To Spell
```

Fuente: elaboración propia, empleando captura de pantalla.

En la segunda sección, se crean dos archivos, el primero será utilizado para la base de datos de respaldo, que quedará guardado en la *Raspberry Pi* y el nombre está basado según la fecha diaria, es decir, creará archivos nuevos cada día. En el segundo archivo guardaremos únicamente la variable de interés en este caso únicamente los valores de presión. Seguidamente se procede a abrir los archivos para ir agregando información y por último se presentan las variables medidas, temperatura y presión. Ver figura 85.

Figura 85. Código sensor BMP280 parte 2

```
# Abrir archivo y agregar
# Nombre de archivo basado en fecha actual
filename = '/home/pi/data/wx_stn/BMP280/bmp280_%4d%02d%02d.dat' % (now.year, now.month, now.day)
filename2 = '/home/pi/Desktop/Concatenado/Presion.txt'

#Abrir archivo
file = open(filename, 'a')
file2 = open(filename2, 'a')

try:
    # Variables medidas o derivadas
    tempC = temperature
    tempF = temperature*1.8+32
    slp = (pressure/100.0)*pow(1-(0.0065*altitude)/(tempC+0.0065*altitude+273.15),-5.257)
    station_pres = pressure/100.0
    slp_hg = slp/33.86389
```

Fuente: elaboración propia, empleando captura de pantalla.

En la tercera parte se escriben los datos en los archivos creados y se procede a cerrar los archivos hasta el nuevo registro. Ver figura 86.

Figura 86. Código sensor BMP280 parte 3

```
# Escribir archivo
file.write("%02d %02d %4d %02d %02d %6.2f %6.2f %6.2f %7.2f %5.2f %7.2f\n" % (now.month, now.day, now.year, now.hour, now.minute, tempC, tempF, station_pres))
file2.write("%6.2f %7.2f %5.2f %7.2f\n" % (station_pres, slp, slp_hg, altitude))

finally:
    if file is not None:
        file.close()
    file2.close()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.2. Sensor de humedad HTU21D

En la primera parte del código mostrado, ver figura 87, se inicializa el programa con la línea *shebang* para la creación del *script*, la importación de librerías necesarias, así como la declaración del sensor como esclavo para el uso de I2C y la creación de distintas clases y procesos (función) para la utilización de I2C y la toma de datos.

Figura 87. Código sensor HTU21D parte 1

```
GNU nano 2.2.6 File: htu21d_read_2file_2web.py
#!/usr/bin/python
# Código para contor de sensor HTU21D temp y humedad. Escribir observaciones en disco

# import librerias necesarias
import sys, struct, array, time, io, os, fcntl, datetime

import requests

I2C_SLAVE=0x0703
HTU21D_ADDR = 0x40
CMD_READ_TEMP_HOLD = "\xE3"
CMD_READ_HUM_HOLD = "\xE5"
CMD_READ_TEMP_NOHOLD = "\xF3"
CMD_READ_HUM_NOHOLD = "\xF5"
CMD_WRITE_USER_REG = "\xE6"
CMD_READ_USER_REG = "\xE7"
CMD_SOFT_RESET= "\xFE"

class i2c(object):
    def __init__(self, device, bus):
        self.fr = io.open("/dev/i2c-"+str(bus), "rb", buffering=0)
        self.fw = io.open("/dev/i2c-"+str(bus), "wb", buffering=0)

        # Se setea la direccion del dispositivo
        fcntl.ioctl(self.fr, I2C_SLAVE, device)
        fcntl.ioctl(self.fw, I2C_SLAVE, device)

    def write(self, bytes):
        self.fw.write(bytes)

    def read(self, bytes):
        return self.fr.read(bytes)

    def close(self):
        self.fw.close()
        self.fr.close()

class HTU21D(object):
    def __init__(self):
        self.dev = i2c(HTU21D_ADDR, 1) #HTU21D 0x40, bus 1
        self.dev.write(CMD_SOFT_RESET) #soft reset
        time.sleep(.1)

    def ctemp(self, sensorTemp):
        tSensorTemp = sensorTemp / 65536.0
        return -46.85 + (175.72 * tSensorTemp)

    def chumid(self, sensorHumid):
        tSensorHumid = sensorHumid / 65536.0
        return -6.0 + (125.0 * tSensorHumid)

    def crc8check(self, value):
        # Importado desde Sparkfun Arduino HTU21D Library: https://github.com/sparkfun/HTU21D_Breakout
        remainder = ( ( value[0] << 8 ) + value[1] ) << 8
        remainder |= value[2]
```

Fuente: elaboración propia, empleando captura de pantalla.

En la segunda y tercera parte, ver figuras 89 y 90, se realiza la utilización de un polinomio de grado 8, y un divisor de polinomio desplazado al extremo izquierdo a tres bytes para la obtención de datos y la aplicación de la librería del sensor. También se muestra el proceso de impresión de datos en consola y la toma de fecha actual.

Figura 88. Código sensor HTU21D parte 2

```
def crc8check(self, value):
# Importado desde Sparkfun Arduino HTU21D Library: https://github.com/sparkfun/HTU21D_Breakout
remainder = ( ( value[0] << 8 ) + value[1] ) << 8
remainder |= value[2]
```

Fuente: elaboración propia, empleando captura de pantalla.

Figura 89. Código sensor HTU21D parte 3

```
GNU nano 2.2.6 File: htu21d_read_2file_2web.py

# Polinomio = 0x0131 = x^8 + x^5 + x^4 + 1
# divisor = 0x988000 es el 0x0131 polinomio desplazado al extremo izquierdo a tres bytes
divsor = 0x988000

for i in range(0, 16):
    if( remainder & 1 << (23 - i) ):
        remainder ^= divsor
        divsor = divsor >> 1

if remainder == 0:
    return True
else:
    return False

def read_temperature(self):
self.dev.write(CMD_READ_TEMP_NOHOLD) #medicion de temperatura
time.sleep(.1)

data = self.dev.read(3)
buf = array.array('B', data)

if self.crc8check(buf):
    temp = (buf[0] << 8 | buf [1]) & 0xFFFC
    return self.ctemp(temp)
else:
    return -255

def read_humidity(self):
self.dev.write(CMD_READ_HUM_NOHOLD) #medicion de humedad
time.sleep(.1)

data = self.dev.read(3)
buf = array.array('B', data)

if self.crc8check(buf):
    humid = (buf[0] << 8 | buf [1]) & 0xFFFC
    return self.chumid(humid)
else:
    return -255

if __name__ == "__main__":
obj = HTU21D()
tempC = obj.read_temperature()
tempF = obj.read_temperature()*1.8+32
rh = obj.read_humidity()
print "HTU21D Sensor"
print "Temperature:", '{:.2f}'.format(obj.read_temperature()), "degC"
print "Temperature:", '{:.2f}'.format((obj.read_temperature()*1.8+32)), "degF"
print "Humidity:", '{:.0f}'.format(obj.read_humidity()), "%"

# Obener la fecha actual
now = datetime.datetime.now()

# Agregar la seccion para escribir el archivo
# Abrir y editar el archivo
# Nombre de archivo basado en fecha
filename = '/home/pi/data/wx_stn/HTU21D/htu21d_%4d%02d%02d.dat' % (now.year, now.month, now.day)
```

Fuente: elaboración propia, empleando captura de pantalla.

En la cuarta y última parte del código se muestra la creación de archivo, el proceso para abrir el archivo, impresión de datos en pantalla tomados del objeto mostrado en la parte 3, el proceso de escritura de datos en los archivos y se cierra el archivo hasta la nueva toma de datos. Ver figura 91.

Figura 90. Código sensor HTU21D parte 4

```
# Agregar la seccion para escribir el archivo
# Abrir y editar el archivo
# Nombre de archivo basado en fecha
filename = '/home/pi/data/wx_stn/HTU21D/htu21d_%4d%02d%02d.dat' % (now.year, now.month, now.day)
filename2 = '/home/pi/Desktop/Concatenado/Humedad.txt'

# Abrir archivo
file = open(filename, 'a')
file2 = open(filename2, 'a')

# Imprimir en pantalla
# print "Abriendo archivo de datos: \n", filename

try:
    tempC = obj.read_temperature()
    tempF = obj.read_temperature()*1.8+32
    rh = obj.read_humidity()

# Escribir en el archivo
file.write('%02d %02d %4d %02d %02d %5.1f %5.1f %4.1f\n' % (now.month, now.day, now.year, now.hour, now.minute, tempC, tempF, rh))
file2.write('%4.1f\n' % (rh))

finally:
    if file is not None:
        file.close()
    file2.close()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.3. Sensor de temperatura MCP9808

En la figura 92, se muestra el código para el sensor, donde se inicializa el programa con la línea de *Shebang* para la interpretación del *script*, la importación de librerías, la creación de la función para convertir la variable de grados centígrados a Fahrenheit, la definición de variables, la inicialización del sensor, obtención de datos, el método de impresión de datos en consola, se procede a obtener la fecha actual para la creación del archivo, el manejo de archivos para crear, abrir, escribir y cerrar el archivo hasta la nueva toma de datos.

Figura 91. Código sensor MCP9808

```
GNU nano 2.2.6 File: mcp9808_measure_2file_2web.py
#!/usr/bin/python
# Codigo para controlar el sensor MCP9808 de temperatura. Escribir observaciones en el disco

#Import librerias a utilizar
import sys, struct, array, time, io, os, fcntl, datetime
import Adafruit_MCP9808.MCP9808 as MCP9808
import requests

# Funcion para convertir de celsius a fahrenheit.
def c_to_f(c):
    return c * 9.0 / 5.0 + 32.0

# Definir la variable del sensor
sensor = MCP9808.MCP9808()

# Inicializar comunicacion con el sensor
sensor.begin()

# Obtener el dat
tempC = sensor.readTempC()
tempF = c_to_f(tempC)

# Imprimir en la pantalla
print "MCP9808 Sensor"
print "Temperature : " + '{:.2f}'.format(tempC) + " degC"
print "Temperature : " + '{:.2f}'.format(tempF) + " degF"

# Obtener fecha actual
now = datetime.datetime.now()

# Crear nombre de archivo basado en fecha
filename = '/home/pi/data/wx_stn/MCP9808/mcp9808_%.4d%.02d%.02d.dat' % (now.year, now.month, now.day)
filename2 = '/home/pi/Desktop/Concatenado/Temperatura.txt'

# Abrir documento - Agregar dato si ya existe
file = open(filename, 'a')
file2 = open(filename2, 'a')

# Escribir datos en el archivo
try:
    file.write('%02d %02d %4d %02d %02d %5.1f %5.1f\n' % (now.month, now.day, now.year, now.hour, now.minute, tempC, tempF))
    file2.write('%5.1f, %5.1f\n' % (tempC, tempF))
# Cerrar archivo
finally:
    if file is not None:
        file.close()
    file2.close()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.4. Sensor Littelfuse 55300-00-002-A

En la primera parte del código mostrado en la figura 93, se muestra la línea de *Shebang* para el uso del *script*, la importación de librerías, el tiempo de procesamiento, la función para la dirección de viento, el valor inicial de conteo para el tiempo y dirección de viento, la toma de fecha actual para el manejo de archivos y el ciclo para la lectura de datos del sensor de viento.

Figura 92. Código sensor *Littelfuse 55300-00-002-A* parte 1

```
GNU nano 2.2.6 File: winddir_1minAvg_2file_2web.py
#!/usr/bin/python3
# Código para control del sensor de dirección de viento. Escribe observaciones en disco
# calcula una media geométrica de 1 minuto.
# Import librerías a utilizar

import spidev, time, os, glob, csv, datetime

spi = spidev.SpiDev()
spi.open(0,0)

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(13, GPIO.OUT)

import requests

# Sleep time, es necesario tener en cuenta el procesamiento de ~0.03 segundos
SLEEP_TIME = 0.97

# Channel 0 es para dirección de viento
def analog_read(channel):
    if ((channel > 1) or (channel < 0)):
        return -1

    r = spi.xfer2([1, (2 + channel) << 6, 0])
    adc_out = ((r[1]&31) << 6) + (r[2]>>2)
    return adc_out

# Toma de fecha actual
now_file = datetime.datetime.now()

# Conteo de viento inicial
wnd_cnt = 1
time_cnt = 1

# Lectura de viento desde el sensor
while True:
    # Sensor de viento esta en channel 0
    ch = 0

    # lectura del canal
    reading = analog_read(ch)

    # Entrada min/max de ADC tomada de calibración
    ADC_min = 103
    ADC_max = 920
    ADC_range = ADC_max - ADC_min

    # Conversión de grados desde el norte
    if (reading < ADC_min):
        wnddir = 0
    elif (reading > ADC_max):
        wnddir = 360
    else:
        # Toma de ADC a cero relativo
        ADC_rel = reading - ADC_min
```

Fuente: elaboración propia, empleando captura de pantalla.

En la segunda parte del código se muestra la conversión de grados desde el norte, la toma de un cero relativo, el registro del primer ángulo y la asignación del siguiente valor, así como la incrementación de conteo en 1.

Figura 93. Código sensor *Littelfuse 55300-00-002-A* parte 2

```
# Conversion de grados desde el norte
if (reading < ADC_min):
    wnddir = 0
elif (reading > ADC_max):
    wnddir = 360
else:
    # Toma de ADC a cero relativo
    ADC_rel = reading - ADC_min
    wnddir = ADC_rel*360.0/ADC_range

# Registra el primer angulo
if(time_cnt == 1):
    wnddir_sum = wnddir
    wnddir_prev = wnddir
    time_cnt = time_cnt + 1
elif (time_cnt < 60):
    delta_wnd = wnddir - wnddir_prev
    # Asigna el siguiente valor
    if( delta_wnd < -180.0):
        wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd + 360.0)
        wnddir_prev = wnddir_prev + delta_wnd + 360.0
        wnd_cnt = wnd_cnt + 1
    elif (abs(delta_wnd) < 180.0):
        wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd)
        wnddir_prev = wnddir_prev + delta_wnd
        wnd_cnt = wnd_cnt + 1
    elif (delta_wnd > 180):
        wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd - 360.0)
        wnddir_prev = wnddir_prev + delta_wnd - 360.0
        wnd_cnt = wnd_cnt + 1
    else:
        print("La diferencia de grados es 180, es indefinido y no se utiliza")

# Incremente el tiempo de conte en 1
time_cnt = time_cnt + 1
```

Fuente: elaboración propia, empleando captura de pantalla.

En la tercera parte, ver figura 95, se resuelve el conflicto para la dirección de 360 ° o 0 °, se realiza la lectura de la fecha actual para el manejo de archivos, se crea el archivo basado en fecha, se abre y escriben datos recolectados en el archivo.

Figura 94. Código sensor *Littelfuse 55300-00-002-A* parte 3

```
else:
    if( wnd_cnt > 1):
        wnddir_avg = wnddir_sum/wnd_cnt
        # Resuelve problema de dirección - dir > 360 o < 0.0
        if(wnddir_avg > 360.0):
            wnddir_avg = 360.0
        if(wnddir_avg < 0.0):
            wnddir_avg = 0.0
    else:
        wnddir_avg = -999.0

# Toma de fecha actual para archivo
now = datetime.datetime.now()

# Nombre de archivo basado en fecha
filename = "/home/pi/data/wx_stn/wnddir/wnddir_%.4d%.02d%.02d.dat" % (now.year, now.month, now.day)
filename2 = "/home/pi/Desktop/Concatenado/Wnddir.txt"

# Imprime nombre de archivo en pantalla
# print(filename)

# Abre el archivo de dirección de viento
file = open(filename, "a")
file2 = open(filename2, "a")

# Escribir en archivo y agregar al final del archivo
file.write("%.02d %.02d %.4d %.02d %.02d %.02d %.5d %.5f %.5f\n" % (now.month, now.day, now.year, now.hour, now.minute, now.second, reading, wnddir, w$
file2.write("%.5d, %.5f, %.5f\n" % (reading, wnddir, wnddir_avg))
```

Fuente: elaboración propia, empleando captura de pantalla.

En la cuarta y última parte del código, se muestra el manejo de archivos, donde se realiza la apertura del mismo, la escritura de datos en la última línea y los procesos de cerrar archivos hasta la próxima lectura, se muestran los datos en consola y el restablecimiento de tiempo para el próximo minuto. Ver figura 96.

Figura 95. Código sensor *Littelfuse 55300-00-002-A* parte 4

```
# Abre el archivo de dirección de viento
file = open(filename, "a")
file2 = open(filename2, "a")

# Escribir en archivo y agregar al final del archivo
file.write("%.02d %.02d %.4d %.02d %.02d %.02d %.5d %.5f %.5f\n" % (now.month, now.day, now.year, now.hour, now.minute, now.second, reading, wnddir, w$
file2.write("%.5d, %.5f, %.5f\n" % (reading, wnddir, wnddir_avg))

# Escribir en pantalla
line = "%.02d %.02d %.4d %.02d %.02d %.02d %.5d %.5f %.5f" % (now.month, now.day, now.year, now.hour, now.minute, now.second, reading, wnddir, wnddir_5
# Imprimir en pantalla
print(line)

# Descarga de datos a archivo
file.flush()
file2.flush()

# Cerrar archivo hasta la próxima iteración
file.close()
file2.close()

# Reestablecer el tiempo y los recuentos de viento para el próximo minuto
wnd_cnt = 1
time_cnt = 1

# Sleep por un minuto (incluyendo en segundo actual)
time.sleep(SLEEP_TIME)

# Cerrar el archivo
file.close()
file2.close()
GPIO.cleanup()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.5. Sensor SS451A

En esta sección se realizará la explicación de los códigos fuentes para la toma de datos de velocidad del viento y lluvia, puesto que la toma de datos se realiza mediante el sensor SS451A.

5.3.5.1. Velocidad de viento

En la primera parte del código, se inicia con la línea de *Shebang* para la utilización del *script*, la importación de librerías a utilizar, se declara la variable del contador del sensor, se declara el tiempo de muestro y tiempo de espera, el factor de calibración para la velocidad del viento, se realiza la declaración del offset, ya que se considera el material con el cual está realizada las cazoletas, se crea la función para cada lectura del sensor. Ver figura 97.

Figura 96. Código velocidad de viento sensor SS451A parte 1

```
#!/usr/bin/python3
# Código para controlar el sensor de velocidad del viento. Escribe la observacion en el disco

# importar librerias
import RPi.GPIO as GPIO
import time, datetime
import requests

# Variable de contador del sensor
COUNT = 1

# Numero de sensor en anemometro
SENSOR_NUM = 2

# Tiempo de muestra en segundos
SAMPLE = 58.5

# Sleep time
SLEEP_TIME = 60

# Factor de calibración de la velocidad del viento
CAL_Factor = 2.64 #(3.14/1.19) - Factor original de calibracion

# Factor de compensación para diferentes materiales (PLA)
#OFFSET = 1.43
# Need to have an offset of 0.0
OFFSET = 0.0

SCALE = CAL_Factor*(2*3.14156*0.079)/(SENSOR_NUM*SAMPLE) # Velocidad de viento en m/s

# Identificando el pin GPIO para el sensor de viento
PIN = 22

GPIO.setmode(GPIO.BCM)
GPIO.setup(PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Variable para realizar un seguimiento de la cantidad de rotaciones
wind = 0

# Funcion de devolucion de llamada para cada lectura del sensor de viento
def cb(channel):
    global wind
    wind = wind + COUNT

# Registrar la devolucion de llamada para interrupciones de pin
GPIO.add_event_detect(PIN, GPIO.FALLING, callback=cb, bouncetime=1)

# Obtener la hora actual
now_file = datetime.datetime.now()
```

Fuente: elaboración propia, empleando captura de pantalla.

En la figura 97, se observa el ciclo para mostrar los datos, la toma de fecha actual para el registro de archivos, el uso de archivos para crear, abrir y escribir datos en la última línea del archivo.

Figura 97. **Código velocidad de viento sensor SS451A parte 2**

```
# Mostrar y escribir datos en un archivo y enviarlos
while True:
    if wind > 0:
        wind_spd = wind*SCALE+OFFSET
    else:
        wind_spd = 0.0

    #Obtener la hora actual para cada registro y nombre de archivo
    now = datetime.datetime.now()

    # Abre el archivo y agregar al archivo del día actual
    # Crear nombre de archivo definido por fecha
    filename = "/home/pi/data/wx_stn/windspd/windspd_%d%02d%02d.dat" % (now.year, now.month, now.day)
    filename2 = "/home/pi/Desktop/Concatenado/Windspeed.txt"

    # Mostrar el nombre del archivo
    # print(filename)

    # Abrir el archivo de velocidad de viento
    file = open(filename, "a")
    file2 = open(filename2, "a")

    # Escribir datos en el archivo
    file.write("%02d %02d %4d %02d %02d %4.2f\n" % (now.month, now.day, now.year, now.hour, now.minute, now.second, wind_spd))
    file2.write("%4.2f\n" % (wind_spd))
```

Fuente: elaboración propia, empleando captura de pantalla.

En la tercera parte, ver figura 99, se describe la visualización de datos en terminal, la escritura de datos al archivo en tiempo real, el proceso para cerrar el archivo y poder interactuar hasta el siguiente registro.

Figura 98. **Código velocidad de viento sensor SS451A parte 3**

```
# Escribir datos en la pantalla
line = "%02d %02d %4d %02d %02d %4.2f" % (now.month, now.day, now.year, now.hour, now.minute, now.second, wind_spd)

# Imprimir en la pantalla
print(line)

#flush - Escribe en archivo en tiempo real
file.flush()
file2.flush()

# Cerrar el archivo para la proxima iteracion
file.close()
file2.close()

wind = 0

# Sleep durante 60 segundos, incluido el segundo actual
time.sleep(SLEEP_TIME)

# Cerrar el archivo de registro
file.close()
file2.close()
GPIO.cleanup()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.5.2. Precipitación

En la primera parte del código mostrado en la figura 100, se inicializa con la línea de *Shebang* para la utilización de *script*, la importación de librerías, la calibración de 0,2 mm de lluvia para cada muestreo obtenido, la identificación del GPIO utilizado, el tiempo entre cada medición, la inicialización de dato de precipitación, la definición de la función para la devolución de llamada a cada cuchara del *tipping bucket*, el registro de llamado en las interrupciones hacia el GPIO y la obtención de la fecha actual.

Figura 99. Código precipitación sensor SS451A parte 1

```
#!/usr/bin/python3
#Codigo para controlar el tipping bucket pluviometro. Escribir observaciones en disco

#importar librerias necesarias
import RPi.GPIO as GPIO
import time, datetime

import requests

# Acumulado de lluvia en mm por punta de cuchara
CALIBRATION = 0.2

# Identificando GPIO para sensor de lluvia
PIN = 23

# Sleep time entre mediciones
SLEEP_TIME = 58.5

GPIO.setmode(GPIO.BCM)
GPIO.setup(PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Variable para realizar un seguimiento de la cantidad de lluvia
rain = 0

# Funcion de devolucion de llamada para cada punta de cubo
def cb(channel):
    global rain
    rain = rain + CALIBRATION

# Registrar la devolucion de llamada para interrupciones de pin
GPIO.add_event_detect(PIN, GPIO.FALLING, callback=cb, bouncetime=300)

# Obtener fecha actual
now_file = datetime.datetime.now()
```

Fuente: elaboración propia, empleando captura de pantalla.

En la segunda parte, se muestra el proceso de manejo de datos indicando desde la creación de archivo, el proceso de escritura de datos en archivo y consola, el tiempo de espera hasta la siguiente medición y el proceso de cerrado de archivo. Ver figura 100.

Figura 100. Código precipitación sensor SS451A parte 2

```
# Escribir archivo
while True:
    # Obtener fecha y hora actual para el registro y nombre de archivo
    now = datetime.datetime.now()

    # Escribir en archivos diarios, usando fecha actual para el archivo
    # Asignar nombre de archivo basado en fecha actual
    filename = "/home/pi/data/wx_stn/rain/rain_tip_%4d%02d%02d.dat" % (now.year, now.month, now.day)
    filename2 = "/home/pi/Desktop/Concatenado/rain.txt"

    # Mostrar nombre de archivo en pantalla
    # print(filename)

    # Abrir archivo
    file = open(filename, "a")
    file2 = open(filename2, "a")

    # Escribir en archivo
    file.write("%02d %02d %4d %02d %02d %02d %4.2f\n" % (now.month, now.day, now.year, now.hour, now.minute, now.second, rain))
    file2.write("%4.2f\n" % (rain))
    # Crear línea para mostrar en pantalla
    line = "%02d %02d %4d %02d %02d %02d %4.2f" % (now.month, now.day, now.year, now.hour, now.minute, now.second, rain)

    # Imprimir en pantalla
    print(line)

    # Descarga de archivos y enviar datos al archivo
    file.flush()
    file2.flush()

    # Cerrar archivo hasta el proximo minuto
    file.close()
    file2.close()

    rain = 0

    # Sleep hasta la siguiente medicion
    time.sleep(SLEEP_TIME)

# Cerrar archivo de datos
try:
    file.close()
    file2.close()
except:
    pass

GPIO.cleanup()
```

Fuente: elaboración propia, empleando captura de pantalla.

5.3.6. Sensor de intensidad lumínica SI1145

En el código mostrado del sensor de intensidad lumínica, se inicializa con la línea de *Shebang* para el uso de *script*, se importan las librerías a utilizar, se realiza la asignación de la variable para el sensor, el proceso de lectura de datos,

la impresión de datos en la terminal y el proceso de manejo de archivos que han sido descritos en los sensores anteriores. Ver figura 101.

Figura 101. **Sensor de intensidad lumínica SI1145**

```
GNU nano 2.2.6 File: si1145_read_2file_2web.py
#!/usr/bin/python
#Codigo para controlar el sensor SI1145. Escribe la observacion en el disco

# Import libreria SI1145
import SI1145.SI1145 as SI1145

# Import librerias necesarias
import sys, struct, array, time, io, os, fcntl, datetime

import requests

# Asignar variable al nombre del sensor
sensor = SI1145.SI1145()

# Ejecutar el codigo
vis = sensor.readVisible()
IR = sensor.readIR()
UV = sensor.readUV()
uvIndex = UV / 100.0
print 'Vis:          ' + str(vis)
print 'IR:           ' + str(IR)
print 'UV Index:     ' + str(uvIndex)

# Get current time
now = datetime.datetime.now()

# Agregar seccion para escribir en el archivo.
# Abre un archivo y agrega al final
filename = '/home/pi/data/wx_stn/SI1145_%d%02d%02d.dat' % (now.year, now.month, now.day)
filename2 = '/home/pi/Desktop/Concatenado/SI.txt'

# Abrir el archivo
file = open(filename, 'a')
file2 = open(filename2, 'a')

# escribir en el archivo
file.write("%02d %02d %4d %02d %02d %10.3f %10.3f %10.3f %10.3f\n" % (now.month, now.day, now.year, now.hour, now.minute, vis, IR, UV, uvIndex))
file2.write("%10.3, %10.3f, %10.3f, %10.3f\n" % (vis, IR, UV, uvIndex))

# Cerrar el archivo
if file is not None:
    file.close()
if file2 is not None:
    file2.close()
```

Fuente: elaboración propia, empleando captura de pantalla.

Figura 102. **Implementación de GPRS a estación meteorológica.**

En esta sección se mostrará el proceso de configuración de GPRS para la transmisión de datos hacia un equipo de INSIVUMEH, dentro de su red local.

INSIVUMEH entre su red de estaciones meteorológicas, hidrológicas y sísmicas, realiza transmisión de datos de distintos métodos, uno de ellos es el uso de GPRS, contratado con un proveedor local.

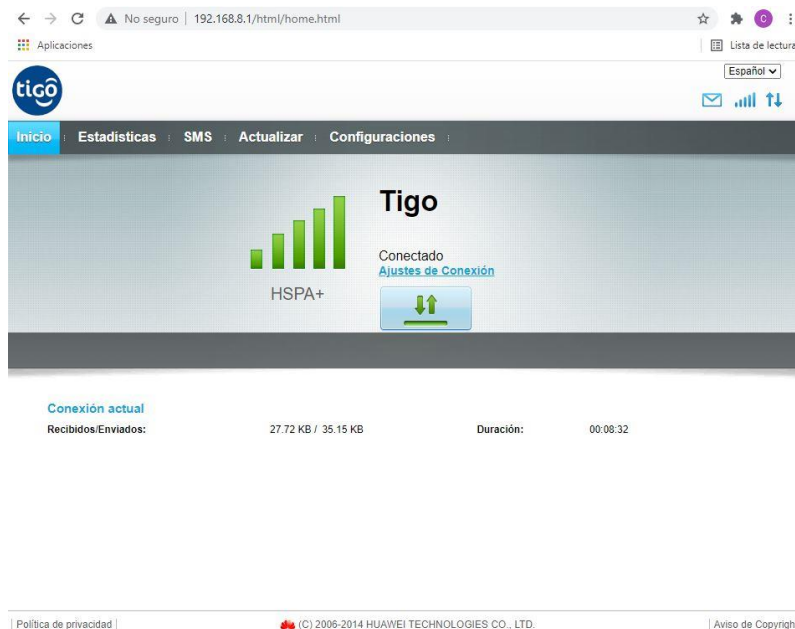
Para este proyecto, INSIVUMEH proporcionó un chip, de una estación meteorológica que actualmente se encuentra fuera de funcionamiento.

5.3.6.1. Configuración de SIM

Para esto utilizaremos un modem USB y el SIM para realizar la asignación de la APN en la configuración inicial de nuestro dispositivo. Esto permitirá realizar el reconocimiento de red necesario para la transmisión de datos hacia un equipo designado por INSIVUMEH para la recolección de datos.

Como primer paso conectamos nuestro modem USB a nuestro ordenador, automáticamente abrirá el navegador web y nos mostrará la página de inicio, en el cual se muestra si existe transferencia de paquetes. Ver figura 103.

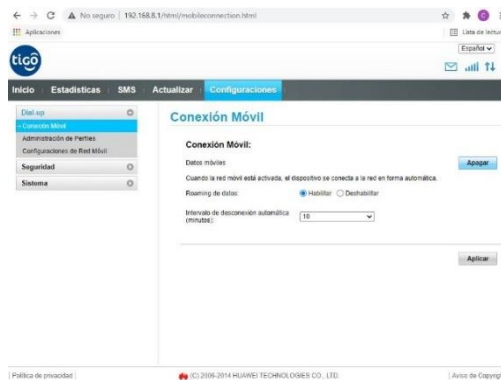
Figura 103. Página de inicio configuración APN



Fuente: elaboración propia, empleando captura de pantalla.

En el menú, presionamos en el apartado de configuraciones. Ver figura 104.

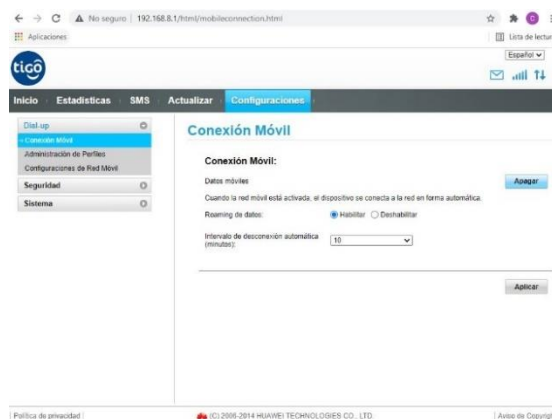
Figura 104. **Página inicial de configuraciones, APN**



Fuente: elaboración propia, empleando captura de pantalla.

Se mostrará un menú secundario, en la sección de Dial-up, presionamos administración de perfiles. Para el proveedor TIGO, el perfil por defecto posee el nombre de BROADBAND TIGO GT, y su APN es broadband.tigo.gt.

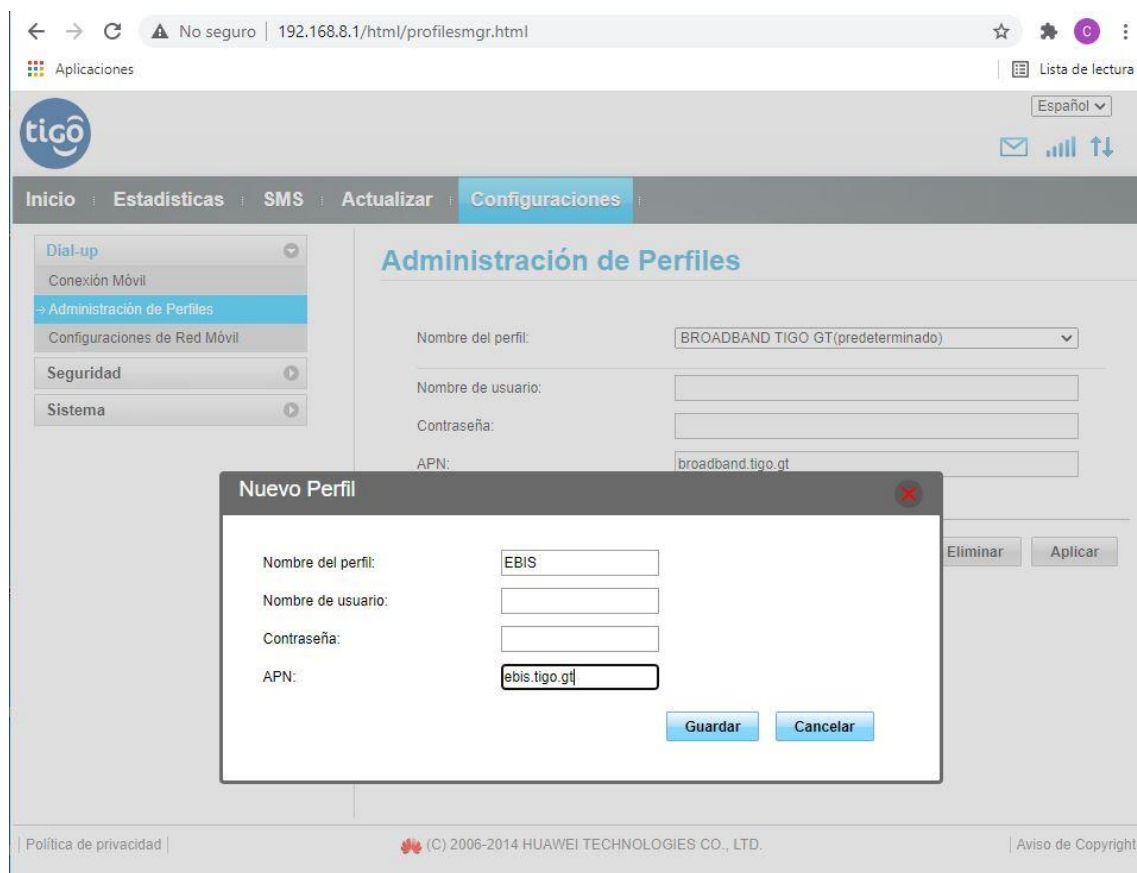
Figura 105. **Administración de perfiles BROADBAND TIGO GT**



Fuente: elaboración propia, empleando captura de pantalla.

Para agregar el nuevo perfil, presionamos en el botón “Nuevo Perfil”, donde se desplegará una ventana donde se pueden agregar los datos del perfil. La APN de INSIVUMEH posee el nombre de EBIS, nombre de usuario se queda vacío, al igual que la contraseña, y la APN es ebis.tigo.gt.

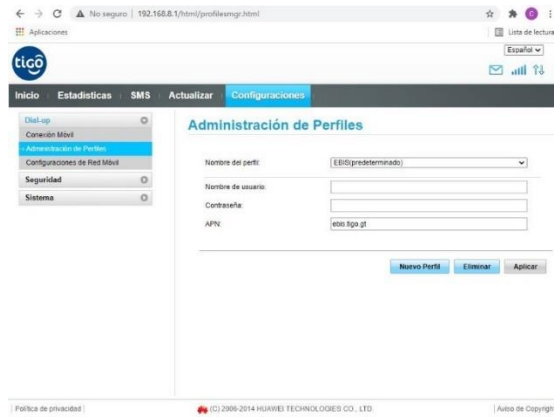
Figura 106. Creación de nuevo perfil



Fuente: elaboración propia, empleando captura de pantalla.

Presionamos el botón de aceptar, la venta desaparecerá, y en la página de administración de perfiles, se habrá configurado el perfil ingresado. Para guardar los cambios se presiona en el botón aplicar. Ver figura 107.

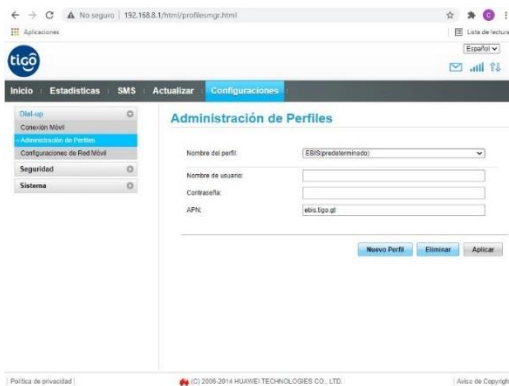
Figura 107. **Aplicar nuevo perfil creado**



Fuente: elaboración propia, empleando captura de pantalla.

En el menú secundario, en la sección de sistema, ingresamos a la información del dispositivo para verificar que nuestro SIM tenga un IP, que pueda ser reconocida por la red de INSIVUMEH. El rango de IP's de INSIVUMEH actualmente está desde 10,251.112,2 a 10,251.112,65, con máscara de subred 255,255.255,0, por motivos de ampliación en la red.

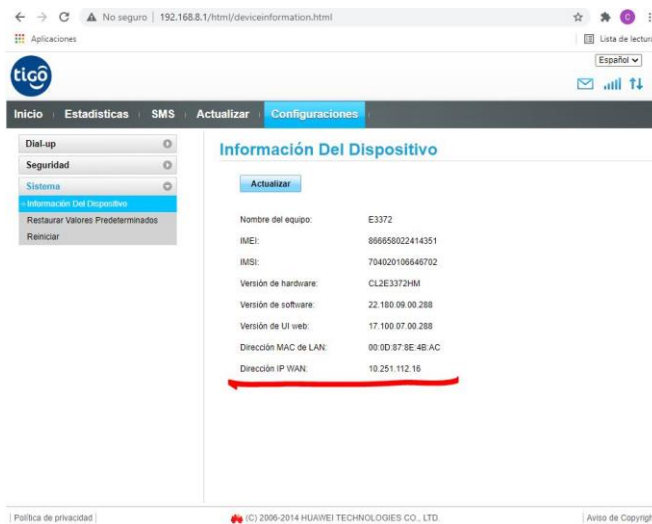
Figura 108. **Búsqueda de información de dispositivo**



Fuente: elaboración propia, empleando captura de pantalla.

En la figura 109, se puede verificar que la IP asociada al SIM se encuentra dentro de los rangos de IP que INSIVUMEH tiene visible en su red.

Figura 109. Información de dispositivo



Fuente: elaboración propia, empleando captura de pantalla.

5.3.6.2. Configuración de GPRS en Raspberry Pi

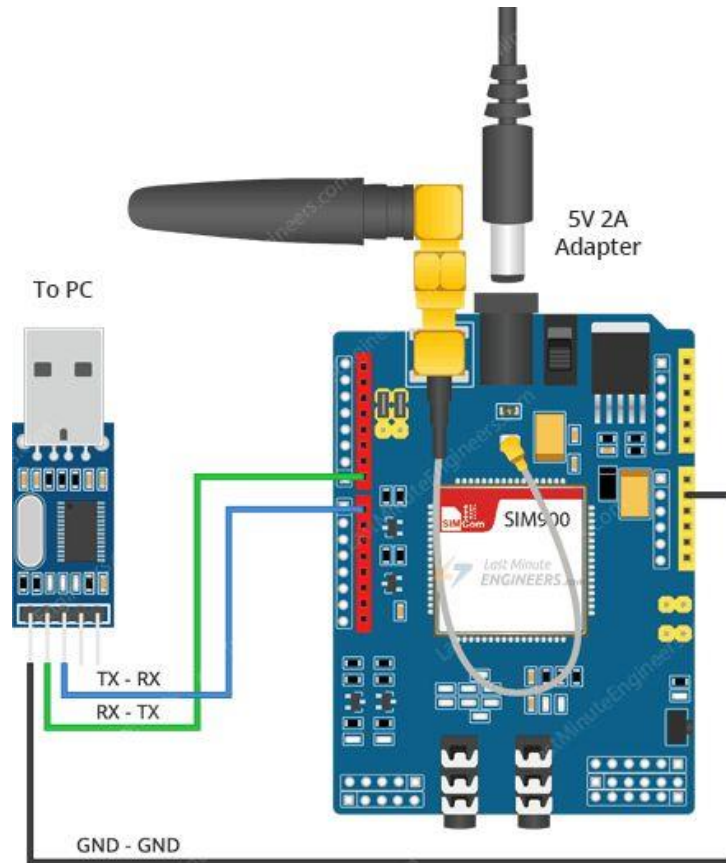
En esta sección se realizará el procedimiento para la implementación de GPRS a la *Raspberry Pi*. Para ello utilizaremos los siguientes materiales.

Tabla XXI. Materiales conexión GPRS a *Raspberry Pi*

Descripción	Unidad	Cantidad
Cable puente hembra a macho	Pieza	3
Módulo USB a TTL	Pieza	1

Fuente: elaboración propia, empleando captura de pantalla.

Figura 110. **Conexión USB – TTL a GRPS**



Fuente: i.pinimg. *Conexión USB – TTL a GRPS.*

<https://i.pinimg.com/564x/a4/05/9f/a4059fcc864e4ac8aa9399c7ac17d240.jpg>. Consulta: 16 de octubre de 2021.

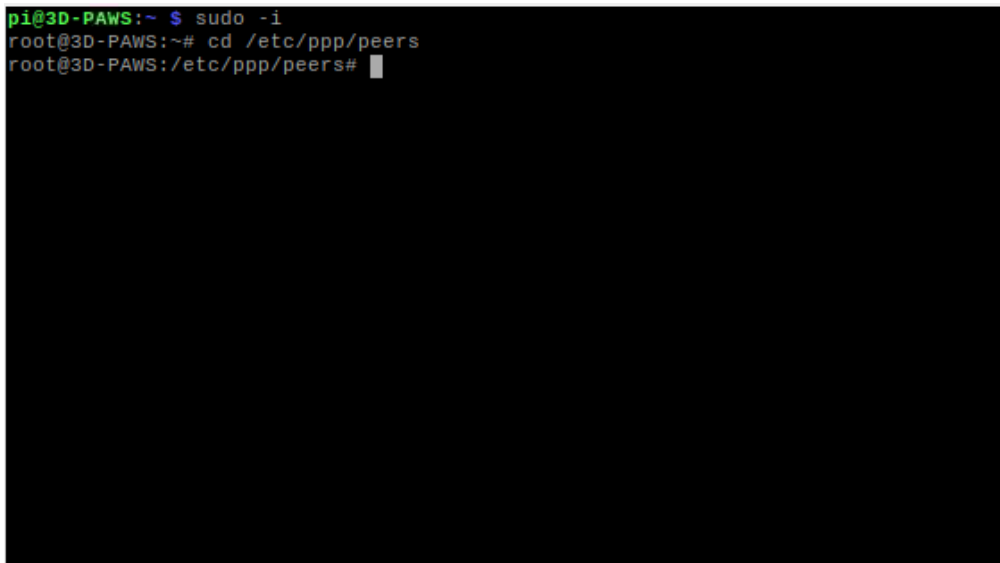
Iniciamos realizando la conexión que se muestra en la figura 110, y procedemos a conecta el USB a nuestra *Raspberry Pi*. Es necesario indicar que se debe proporcionar una fuente de alimentación externa para el módulo GPRS, para obtener un correcto funcionamiento. Al encender el módulo, la tarjeta SIM tardará unos segundos en registrarse en la red.

Seguidamente iniciamos la configuración PPP. El protocolo PPP o punto a punto establece una comunicación de nodo a nodo mediante una interfaz en serie. Se hace uso de esto, mientras accedemos a la conexión de datos en serie de *Raspberry Pi*.

Para la instalación, el primer paso es conectar la *Raspberry Pi* a una conexión de internet, puede ser mediante LAN o WAN; abrimos la terminal de RPi y escribimos “*sudo apt-get update*”. Al finalizar este proceso procedemos ingresando “*sudo apt-get install ppp screen elinks*”.

Después de la instalación, se crea una nueva configuración para el protocolo ppp. Para esta operación, debemos iniciar sesión como usuario *root* ingresando el comando “*sudo su*” o “*sudo -i*” en la terminal y navegar a la siguiente ubicación. “*cd /etc/ppp/peers*”. Ver figura 111.

Figura 111. **Directorio *peers***



```
pi@3D-PAWS:~ $ sudo -i
root@3D-PAWS:~# cd /etc/ppp/peers
root@3D-PAWS:/etc/ppp/peers#
```

Fuente: elaboración propia, empleando captura de pantalla.

Como siguiente paso abrimos el archivo rnet en un editor de texto ejecutando el siguiente comando “nano rnet” y agregamos el siguiente código observado en la figura 112.

Figura 112. Creación archivo rnet



```
GNU nano 3.2                                rnet                                Modificado
#indicamos la apn a conectarnos
connect "/usr/bin/chat -v -f /etc/chatscripts/gprs -T ebis.tigo.gt"
#indicamos el puerto de comunicacion
/dev/ttyUSB0
#definimos el baudrate de funcionamiento
115200
#intenta obtener la dirección IP
usepeerdns
#Utiliza esta conexión como ruta predeterminada
defaultroute
#Hace que PPP vuelva a marcar cuando se pierda la conexión
persist
#no le pide al control remoto que se autentique
noauth
#Sin control de flujo de hardware en el enlace serial con modem
nocrtscts
#sin líneas de control de modem
local

^G Ver ayuda  ^O Guardar  ^W Buscar  ^K Cortar txt ^J Justificar ^C Posición
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Fuente: elaboración propia, empleando captura de pantalla.

El archivo anterior se conecta mediante un *script* de chat. Para ver este archivo ingresamos el siguiente comando “nano /etc/chatscripts/gprs”. Esto abrirá un nuevo archivo, no es necesario editarlo, únicamente si se posee un PIN asignado a la SIM, de lo contrario el archivo mencionado no es necesario editarlo. Este archivo proporciona comandos AT, para realizar la petición y configuración de nuestro SIM y poder ingresar a la red deseada. Ver figura 113.

Figura 113. Archivo gprs

```
GNU nano 3.2 /etc/chatscripts/gprs
# You can use this script unmodified to connect to cellular networks.
# The APN is specified in the peers file as the argument of the -T command
# line option of chat(8).

# For details about the AT commands involved please consult the relevant
# standard: 3GPP TS 27.007 - AT command set for User Equipment (UE).
# (http://www.3gpp.org/ftp/Specs/html-info/27007.htm)

ABORT          BUSY
ABORT          VOICE
ABORT          "NO CARRIER"
ABORT          "NO DIALTONE"
ABORT          "NO DIAL TONE"
ABORT          "NO ANSWER"
ABORT          "DELAYED"
ABORT          "ERROR"

# cease if the modem is not attached to the network yet
ABORT          "+CGATT: 0"

""            AT
TIMEOUT       12
OK            ATH
OK            ATE1

# +CPIN provides the SIM card PIN
#OK          "AT+CPIN=1234"

# +CFUN may allow to signal the handset to limit operations to
# GPRS/EDGE/UMTS/etc to save power, but the arguments are not standard
# except for 1 which means "full functionality".
#OK          AT+CFUN=1

OK            AT+CGDCONT=1,"IP","", "",0,0
OK            ATD*99#
TIMEOUT       22
CONNECT      ""

^G Ver ayuda  ^O Guardar   ^W Buscar    ^R Cortar txt ^J Justificar ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich.^E Reemplazar ^U Pegar txt  ^T Ortografía ^_ Ir a línea  M-E Rehacer
```

Fuente: elaboración propia, empleando captura de pantalla.

En la figura 113, se marca la línea que se debe retirar el símbolo numeral que se encuentra al inicio de la línea y realizar el cambio de PIN asignado a la SIM si fuese necesario.

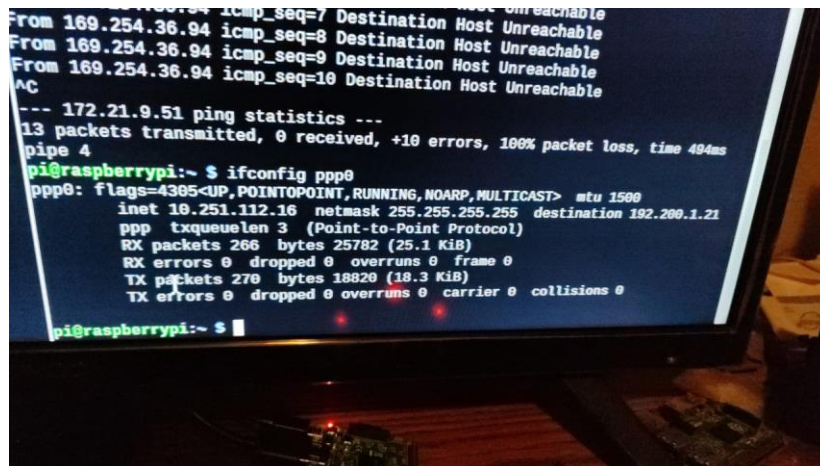
Para iniciar el proceso, deshabilitamos las conexiones LAN y WAN con los siguientes comandos:

- `Sudo ifconfig eth0 down.`
- `Sudo ifconfig wlan0 down.`

Y para iniciar el proceso y detener la conexión ppp, realizamos los siguientes comandos:

- `Sudo pon` (para iniciar).
- `Sudo poff` (para finalizar).
- Para verificar la ip, realizamos el comando `ifconfig ppp0` y nos desplegará la ip de nuestro dispositivo. Ver figura 114.

Figura 114. ***Ifconfig pp0***



```
From 169.254.36.94 icmp_seq=7 Destination Host Unreachable
From 169.254.36.94 icmp_seq=8 Destination Host Unreachable
From 169.254.36.94 icmp_seq=9 Destination Host Unreachable
From 169.254.36.94 icmp_seq=10 Destination Host Unreachable
AC
--- 172.21.9.51 ping statistics ---
13 packets transmitted, 0 received, +10 errors, 100% packet loss, time 494ms
pipe 4
pi@raspberrypi:~$ ifconfig ppp0
ppp0: flags=4385<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.251.112.16 netmask 255.255.255.255 destination 192.200.1.21
ppp txqueuelen 3 (Point-to-Point Protocol)
RX packets 266 bytes 25782 (25.1 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 270 bytes 18820 (18.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pi@raspberrypi:~$
```

Fuente: elaboración propia, empleando captura de pantalla.

Al momento de ejecutar el comando “sudo pon”, se realiza la comunicación al GPRS mediante los comandos AT, en el cual se declara la comunicación TCP

con la siguiente línea: AT+CIPSTART="TCP", "10,251.112,16", "30000", donde se indica el protocolo de comunicación, la IP del dispositivo y el puerto de comunicación.

5.3.7. Base de datos

Para la creación de la base de datos, utilizaremos el uso de XAMPP. Esta es una herramienta de desarrollo que permite probar el desarrollo web basado en PHP en ordenadores, incluso sin el uso de internet.

Los componentes esenciales, de interés para este proyecto, son el uso de Apache, MySQL y PHP.

Apache es un servidor web, de código abierto, usado globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como *software* libre.

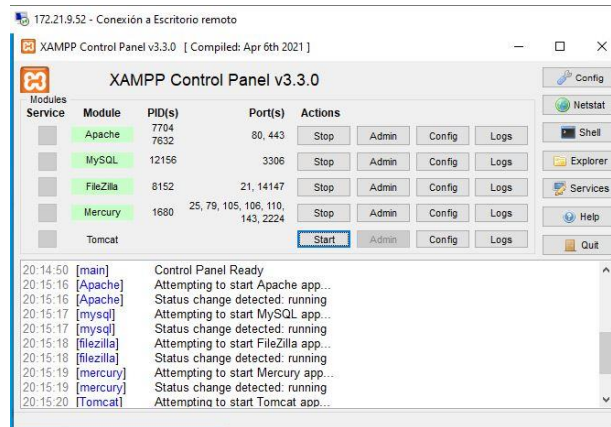
MySQL es conocido por la gestión de base de datos más populares a nivel mundial. En combinación con el servidor Web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web.

PHP es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente a la plataforma y soporta varios sistemas de bases de datos.

Para realizar la descarga de XAMPP, se hace desde su sitio web oficial <https://www.apachefriends.org/es/index.html> y se procede a elegir el sistema operativo de nuestro ordenador, donde será instalada esta aplicación. Realizamos los siguientes pasos:

- Ejecutamos el archivo .exe haciendo doble clic en él, se abrirá un asistente de instalación.
- Los componentes mínimos que instala XAMPP son el servidor Apache y el lenguaje PHP, pero XAMPP también instala otros elementos. En la pantalla de selección de componentes puede elegir la instalación o no de estos componentes, para este proyecto al menos necesitamos agregar MySQL y phpMyAdmin.
- En la siguiente pantalla puede elegir la carpeta de instalación de XAMPP. La carpeta de instalación predeterminada es C:\xampp.
- La siguiente pantalla ofrece información sobre los instaladores de aplicaciones para XAMPP. Para esta ocasión dejamos desmarcada la casilla.
- Una vez elegidas las opciones de instalación en las pantallas anteriores. Hacemos clic en el botón siguiente para comenzar la instalación en el disco duro. El proceso de copia de archivo puede durar varios minutos.
- Una vez terminada la copia de archivos, la pantalla final confirma que XAMPP ha sido instalado. Si deja marcada la casilla, se abrirá el panel de control de XAMPP. Para cerrar el programa de instalación, hacemos clic en el botón finalizar. Para iniciar los procesos, desde la pantalla de control, presionamos los botones “*Start*” de los servicios de nuestro interés. Ver figura 115.

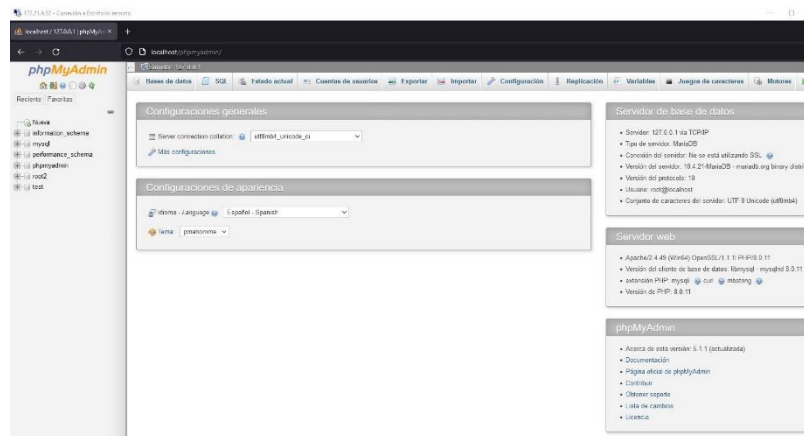
Figura 115. Pantalla de configuración XAMPP



Fuente: elaboración propia, empleando captura de pantalla.

Para ingresar a phpMyAdmin, ingresamos en el navegador web: localhost/phpmyadmin, y se desplegará una ventana como se muestra en la figura 116.

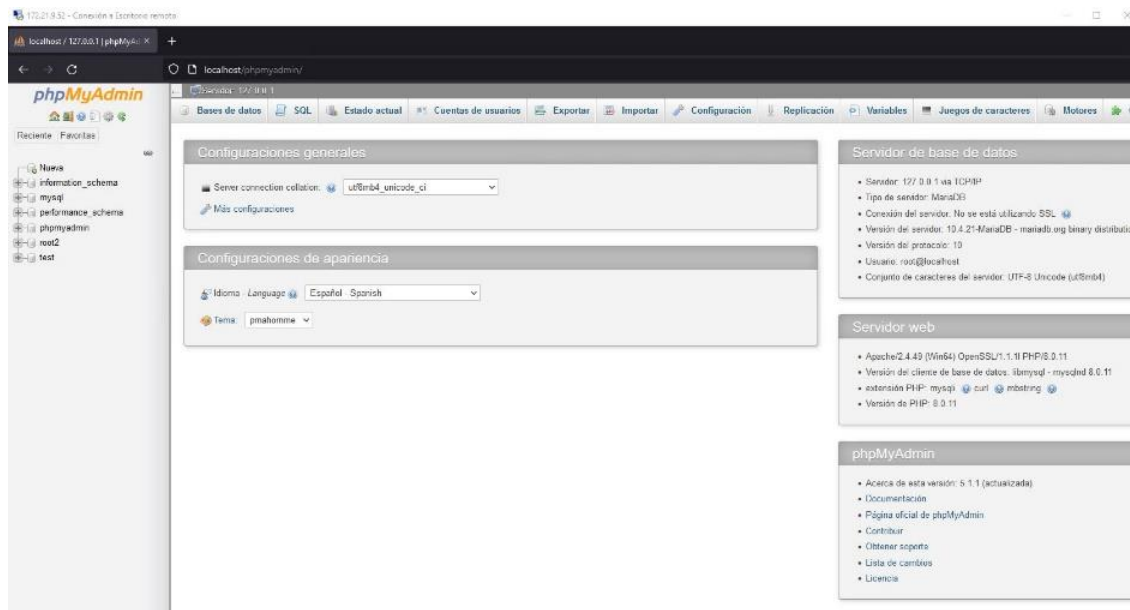
Figura 116. Pantalla de inicio phpMyAdmin



Fuente: elaboración propia, empleando captura de pantalla.

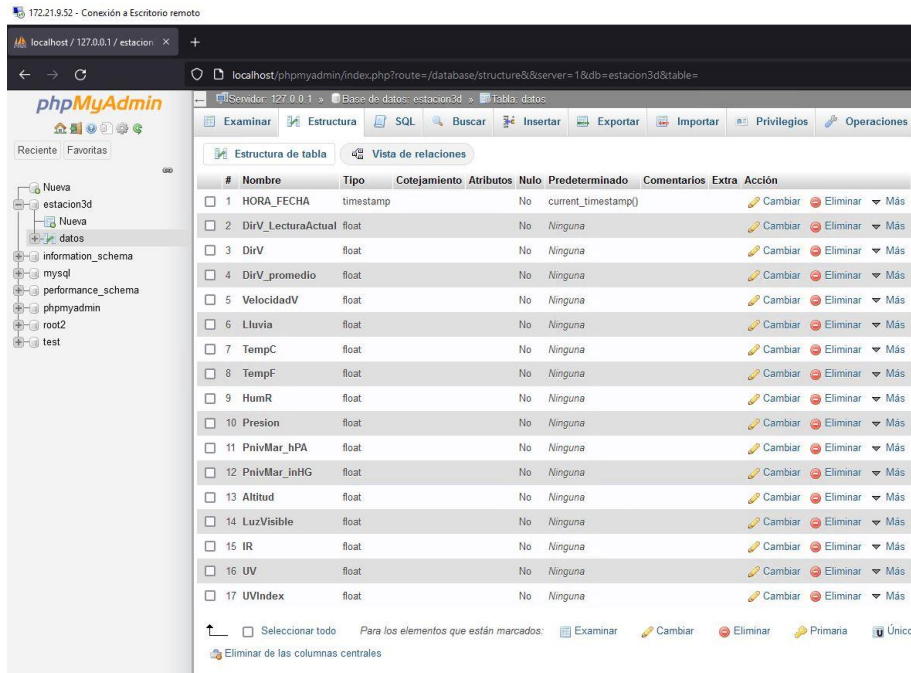
Para la creación de una base de datos, presionamos en la pestaña de Base de Datos, procedemos a nombrar nuestra BD y seguidamente creamos nuestra tabla. Ver figuras 117 y 118.

Figura 117. Creación de tabla datos



Fuente: elaboración propia, empleando captura de pantalla.

Figura 118. Contenido tabla datos



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	HORA_FECHA	timestamp			No	current_timestamp()			Cambiar Eliminar Más
2	DirV_LecturaActual	float			No	Ninguna			Cambiar Eliminar Más
3	DirV	float			No	Ninguna			Cambiar Eliminar Más
4	DirV_promedio	float			No	Ninguna			Cambiar Eliminar Más
5	VelocidadV	float			No	Ninguna			Cambiar Eliminar Más
6	Lluvia	float			No	Ninguna			Cambiar Eliminar Más
7	TempC	float			No	Ninguna			Cambiar Eliminar Más
8	TempF	float			No	Ninguna			Cambiar Eliminar Más
9	HumR	float			No	Ninguna			Cambiar Eliminar Más
10	Presion	float			No	Ninguna			Cambiar Eliminar Más
11	PnivMar_hPA	float			No	Ninguna			Cambiar Eliminar Más
12	PnivMar_inHG	float			No	Ninguna			Cambiar Eliminar Más
13	Altitud	float			No	Ninguna			Cambiar Eliminar Más
14	LuzVisible	float			No	Ninguna			Cambiar Eliminar Más
15	IR	float			No	Ninguna			Cambiar Eliminar Más
16	UV	float			No	Ninguna			Cambiar Eliminar Más
17	UVIndex	float			No	Ninguna			Cambiar Eliminar Más

Fuente: elaboración propia, empleando captura de pantalla.

Para ingresar los datos de forma automática se realizó un programa en *Python* denominado *Conexión.py*, el cual importa las librerías de *PyMySQL*, especifica el host a cuál nos deseamos conectar, el puerto, nombre y contraseña de usuario, el nombre de la base de datos. Además, a cada programa se agregaron líneas de código para realizar el ingreso de los datos a la tabla especificada, por último, en el equipo que recolecta los datos, se pueden realizar gráficos de datos obtenidos por la estación, como comparativos en los valores e luz visible y UV para el censo SI1145.

Para la instalación *PyMySQL*, realizamos los siguientes pasos:

- En la terminal, ejecutamos el comando “*pip install mysqlclient*”.

- El segundo y último comando a ejecutar es “*sudo apt-get install Python-pip Python-dev libmysqlclient-dev*”.

Figura 119. **Conexión a base de datos**

```
#!/usr/bin/python

import pymysql

con = pymysql.connect(
    host = '172.21.9.52',
    port = 3306,
    user = 'root',
    password = 'root',
    db = 'datosinsivumeh',
    charset = 'utf8'
)
cur = con.cursor()
cur.execute("show tables")
print()
```

Fuente: elaboración propia, empleando captura de pantalla.

En la figura 119 se observa el host al cual se conecta la RPi, IP asignada por personal de INSIVUMEH para recibir los datos. El puerto por el cual se realiza la comunicación, nombre de usuario y contraseña al ingresar a phpMyAdmin.

En la figura 122, se aprecia que el siguiente dato agregado a la tabla es la velocidad del viento. Es necesario aclarar, que los espacios se deben dejar los espacios en blanco de los demás datos a insertar.

Al finalizar el proceso de insertar datos al BD, se puede apreciar un resultado final en phpMyAdmin como se muestra en la figura 123.

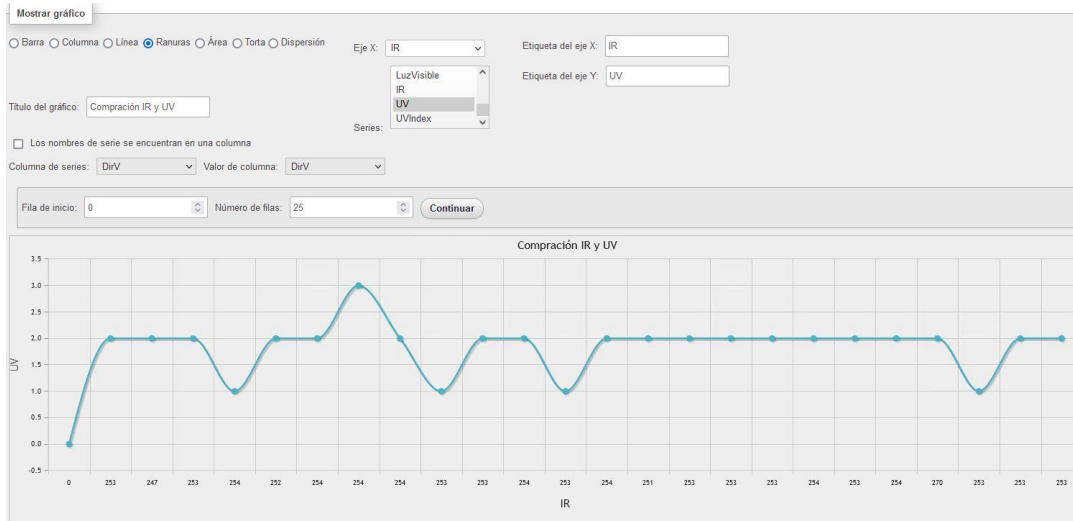
Figura 123. Resultado base de datos

DirV_LecturaActual	DirV	DirV_promedio	VelocidadV	Lluvia	TempC	TempF	HumR	Presion	PnivMar_hPA	PnivMar_inHG	Altitud	LuzVisible	IR	UV	UVIndex
139	15.9	0	3.33	0	17.8	64	62.5	850.67	1005.56	29.69	1451.09	260	253	2	0.02
180	33.9	0	2.82	0	17.8	64	62.5	850.74	1005.57	29.69	1450.41	260	247	2	0.02
765	291.7	280.8	3.19	0	17.8	64	62.4	850.73	1005.59	29.7	1450.61	261	253	2	0.02
220	51.6	360	3.86	0	17.8	64	62.5	850.79	1005.57	29.69	1449.93	259	254	1	0.01
873	339.3	21.5	2.71	0	17.7	63.8	62.6	850.79	1005.59	29.69	1450.13	261	252	2	0.02
837	323.4	332.9	3.54	0	17.7	63.8	62.7	850.88	1005.6	29.7	1449.07	260	254	2	0.02
190	38.3	353.4	2.84	0	17.6	63.7	62.8	850.91	1005.61	29.7	1448.78	262	254	3	0.03
178	33	0	3.7	0	17.8	64	62.9	850.93	1005.61	29.7	1448.59	260	254	2	0.02
238	59.5	28.9	3.49	0	17.7	63.8	63	850.97	1005.62	29.7	1448.21	259	253	1	0.01
118	6.6	0	3.03	0	17.7	63.8	63.1	850.96	1005.64	29.7	1448.3	260	253	2	0.02
894	348.5	0	4.77	0	17.7	63.8	63.1	850.95	1005.63	29.7	1448.4	260	254	2	0.02
860	333.6	1.9	3.68	0	17.7	63.8	63.2	851.02	1005.64	29.7	1447.73	259	253	1	0.01
852	330	357	4.62	0	17.7	63.8	63.2	851.02	1005.63	29.7	1447.73	260	254	2	0.02
152	21.6	10.5	4.83	0	17.7	63.8	63.2	851.01	1005.64	29.7	1447.82	261	251	2	0.02
307	89.9	30.6	2.35	0	17.7	63.8	63.4	851.02	1005.63	29.7	1447.73	260	253	2	0.02
170	29.5	51.5	3.1	0	17.6	63.7	63.5	851.02	1005.65	29.7	1447.73	261	253	2	0.02
114	4.8	0	3.2	0	17.6	63.7	63.5	851.07	1005.66	29.7	1447.25	260	253	2	0.02
804	308.9	0	3.79	0	17.6	63.6	63.8	851.02	1005.67	29.7	1447.73	261	254	2	0.02
856	331.8	0	4.86	0	17.6	63.6	63.7	850.99	1005.69	29.7	1448.02	261	253	2	0.02
205	44.9	335.9	3.86	0	17.6	63.7	63.8	851.1	1005.67	29.7	1446.96	261	254	2	0.02
126	10.1	171.6	4.42	0	17.6	63.6	63.9	851.09	1005.69	29.7	1447.06	261	270	2	0.02
207	45.8	3.4	4.1	0	17.6	63.6	63.9	851.12	1005.7	29.7	1446.77	259	253	1	0.01
185	36.1	11.2	3.58	0	17.6	63.6	64	851.12	1005.71	29.7	1446.77	260	253	2	0.02
159	24.7	0	3.8	0	17.5	63.5	64.1	851.1	1005.72	29.7	1446.96	260	253	2	0.02
224	53.3	0	4.7	0	17.5	63.5	64.2	851.12	1005.73	29.7	1446.77	261	261	2	0.02

Fuente: elaboración propia, empleando captura de pantalla.

El mismo programa permite phpMyAdmin, permite realizar gráficas comparativas, donde el usuario puede seleccionar los ejes X y Y a su conveniencia. En la figura 124, se muestra un comparativo entre los datos registrados de IR y radiación UV.

Figura 124. **Comparativo IR y UV**



Fuente: elaboración propia, empleando captura de pantalla.

Una de las aplicaciones más utilizadas para los datos de estaciones meteorológicas automáticas es compartir datos al sistema de guía de inundaciones repentinas con cobertura global, FFGS por su nombre en inglés *Flash Flood Guidance System with Global Coverage*. FFGS fue diseñado y desarrollado para uso interactivo por pronosticadores meteorológicos e hidrológicos en todo el mundo y es apoyado por la WMO, que es la organización mundial de meteorología.

5.3.8. **Acciones automáticas por tiempos**

En esta sección se mostrará los *scripts* creados para la inicialización de procesos de forma automática de la estación meteorológica impresa en 3D.

Un *script* es un documento que contiene instrucciones, escritas en códigos de programación. El *script* es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador.

Los *scripts* se encargan de cumplir las siguientes funciones:

- Combinar componentes.
- Interactuar con el sistema operativo o con el usuario.
- Controlar un determinado programa o aplicación.
- Configurar o instalar sistemas operaciones.

Algunos lenguajes de programación, usada como *script* son:

- *ActionScript*.
- *JavaScript*.
- Lua.
- PHP.
- *Python*.
- *ShellScript*.
- Ruby.
- VBScript.

Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente *scripts* a una hora o fecha específica, en segundo plano. Los procesos que deben ejecutarse y la hora en la que deben realizarlo se especifican en el fichero Crontab.

Crontab es un simple archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado por el usuario. Crontab verificará la fecha y

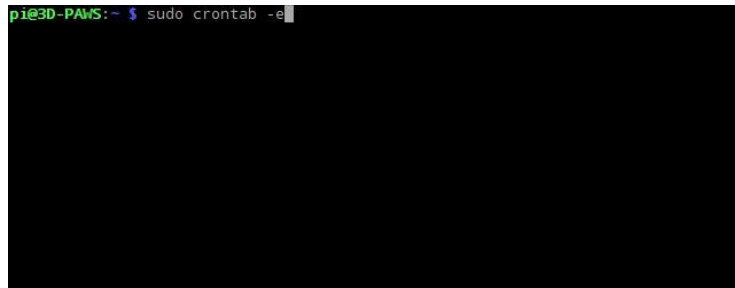
hora en que se debe ejecutar el *script* o el comando. Cada archivo puede tener su propio Crontab.

Para ingresar al fichero de Crontab en la RPi, desde la consola, se ejecuta el comando “sudo crontab -e”. Al ejecutar el comando anterior se desplegará el fichero donde se pueden crear los distintos Crontab. Ver figura 125.

Para este proyecto se realizaron la ejecución de *script* automáticos para los sensores:

- BMP280.
- HTU21D.
- MCP9808.
- SI1145.
- Precipitación.
- Velocidad del viento.
- Dirección del viento
- *Reset* de I2C.
- *Reset* cada 6 horas de RPi (opcional)

Figura 125. **Comando sudo crontab -e**



Fuente: elaboración propia, empleando captura de pantalla.

Figura 126. **Script(s) a ejecutar**

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# 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

## Sensor BMP280 Presion
*/1 * * * * python /home/pi/software/wx_stn/BMP280_pressure_temperature/bmp280_measure_2file_2web.py >> /tmp/bmp280.log 2>&1

## Sensor HTU21D Humedad
*/1 * * * * python /home/pi/software/wx_stn/HTU21D_temperature_humidity/htu21d_read_2file_2web.py >> /tmp/htu21d.log 2>&1

## Sensor MCP9808 Temperatura
*/1 * * * * python /home/pi/software/wx_stn/MCP9808_temperature/mcp9808_measure_2file_2web.py >> /tmp/mcp9808.log 2>&1

## Sensor SI1145 Intensidad Luminica
*/1 * * * * python /home/pi/software/wx_stn/SI1145_UV_light/si1145_read_2file_2web.py >> /tmp/si1145.log 2>&1

## Precipitacion Tipping Bucket
@reboot python /home/pi/software/wx_stn/rain/tipping_bucket_2file_2web.py >> /tmp/rain.log 2>&1

## Direccion del Viento
@reboot python /home/pi/software/wx_stn/winddir/winddir_1minAvg_2file_2web.py >> /tmp/winddir.log 2>&1

## Velocidad del Viento
@reboot python /home/pi/software/wx_stn/windspd/windspd_measure_2file_2web.py >> /tmp/windspd.log 2>&1

## Rebooteo de i2c necesario para cada reboot
@reboot ln -s /dev/i2c-1 /dev/i2c-0

## Opcion de reboot cada 6 horas
#55 2,8,14,20 * * * /sbin/reboot
```

Fuente: elaboración propia, empleando captura de pantalla.

5.4. Incorporación sensor de calidad de Aire a la estructura inicial

En esta sección se hace referencia al EPS desarrollado por el Sr. Araely Ovidio Reyes Carrillo, que lleva por nombre “Diseño e implementación de estación de bajo costo para lectura óptica de partículas PM12 y PM2,5 de calidad de aire para puntos críticos a nivel nacional en el INSIVUMEH”, como un anexo al capítulo cuatro de este proyecto.

Esta incorporación fue solicitada por INSIVUMEH, puesto que en un 95 % de casos las estaciones de calidad de aire necesitan una estación meteorológica asociada con el objetivo de definir las coordenadas provenientes de los contaminantes.

Inicialmente la estructura, elaborada de tubo PVC, estaba sostenida por un tubo de metal galvanizado, en la parte posterior como se muestra en la figura 127.

Figura 127. **Estructura inicial estación meteorológica impresa en 3D**



Fuente: elaboración propia, INSIVUMEH.

A raíz del acople solicitado por la institución se decidió montar la base de la estructura sobre un trípode de acero galvanizado para obtener mayor altura para los sensores de dirección del viento, radiación solar y velocidad de viento, así mismo, poder incorporar el sensor de calidad de aire y una luz estroboscópica como sistema de alerta, al momento de superar los valores normales en las partículas PM12 y PM2,5.

Figura 128. **Incorporación trípode**



Fuente: elaboración propia, INSIVUMEH.

Para la caja que contenía el regulador de voltaje y batería, originalmente se había utilizado una caja para municiones, que por sus características pueden ser utilizadas en la intemperie, pero a raíz del cambio de estructura se realizó el cambio a una caja NEMA 4 con grado de protección IP65 de poliéster, y para el cableado se utilizó unos prensaestopas PG-9. Ver figuras 129 y 130.

Figura 129. **Implementación caja NEMA 4**



Fuente: elaboración propia, INSIVUMEH.

Figura 130. **Prensaestopas PG-9**



Fuente: elaboración propia, INSIVUMEH.

Tabla XXII. **Consumo energético**

Descripción	Cantidad de dispositivos	Consumo máximo	Consumo máximo total	Potencia
Sensor MCP9808	1	0,4 mAh	0,4 mAh	2 mWh
Sensor HTU21D	1	0,5 mAh	0,5 mAh	2,5 mWh
Sensor BMP280	1	1,12 mAh	1,12 mAh	5,6 mWh
<i>Littelfuse 55300-00-02-A</i>	1	16 mAh	16 mAh	0,08 Wh
Sensor SS451A	2	20 mAh	40 mAh	0,2 Wh
Sensor SI1145	1	5,5 mAh	5,5 mAh	0,0275 Wh
<i>Raspberry Pi</i>	2	350 mAh	700 mAh	3,5 Wh
L298N	1	36 mAh	36 mAh	0,432 Wh
Módulo GPRS	1	1 500 mAh	1 500 mAh	7,5 Wh
Módulo de Relay	1	105 mAh	105 mAh	0,525 Wh
Luz estroboscópica	1	60 mAh	60 mAh	0,72 Wh
Sensor NOVA SDS001	1	100 mAh	100 mAh	0,5 Wh
Total			2564,52 mAh	13,4946 Wh

Fuente: elaboración propia, empleando Microsoft Word 365.

Tabla XXIII. **Sistema de alimentación**

Descripción	Cantidad de dispositivos	Valores máximos
Batería recargable 12 V – 18 Amp. Hr.	2	14,4 V – 15 V
Panel solar de 20 W	2	Entrega 17,2 V máximo

Fuente: elaboración propia, empleando Microsoft Word 365.

Para el cálculo de potencia se utilizó la fórmula, descrita a continuación; donde P es potencia, I es corriente y V es voltaje.

$$P = I * V$$

Para el acople de este proyecto, se realizó la conexión de paneles solares en paralelo, de esta forma se mantiene el voltaje o tensión de los paneles solares, mientras que se suma la intensidad de corriente en amperios. Esto se ejemplifica en la figura 131.

Para el cálculo de banco de baterías se aplican las siguientes fórmulas:

$$C_{nom\ banco} = \frac{Energía\ Generada * 12h}{V_{nom} * Ef.\ Descarga * Ef.\ Conversión * Ef.\ inversor}$$

$$NBT = \frac{V_{nom} * C_{nom\ banco}}{V_{nom_{bat}} * C_{nom_{bat}}}$$

$$C_{nom\ banco} = \frac{(13,4946)(12h)}{(12)(0,8)(0,9)(0,75)} = 24,99\ A$$

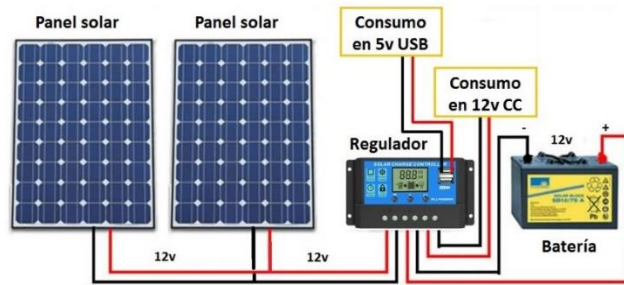
$$NBT = \frac{(12)(24,99)}{(12)(12)} = 2,0825 \approx 2\ baterías$$

Tabla XXIV. Cálculo de autonomía

Descripción	Valor
Energía Generada	13,4946 Wh
Voltaje nominal	12V
Eficiencia de descarga	80 %
Eficiencia de conversión	90 %
Eficiencia de inversor	75 %
Carga nominal de banco (C _{nom banco})	24,99
Carga nominal batería (C _{nom bat})	12 V
Número de Baterías (NBT)	2,083 ≈ 2 baterías

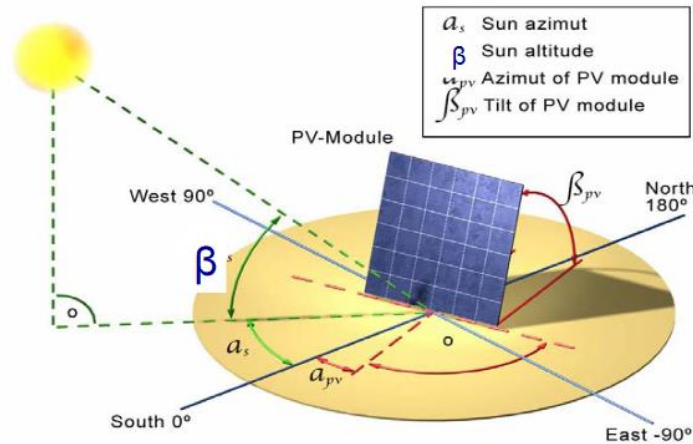
Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 131. **Unidad de alimentación**



Fuente: Genera tu luz. *Generadores solares*. <https://www.generatuluz.com/generar-electricidad/generadores-solares/>. Consulta: 17 de octubre de 2021.

Figura 132. **Inclinación de panel solar**



Fuente: Coa ATC. *Inclinación del panel solar*.

<http://www.coaatc.es/pdf/cte/Comunes%20HE4%20y%20HE5.pdf>. Consulta: 17 de octubre de 2021.

Para realizar el cálculo del ángulo de inclinación del panel solar, se aplica la siguiente fórmula: $\beta = \text{latitud} + 15^\circ$. Donde la latitud de la ubicación de ciudad

Guatemala, sitio donde se encuentra ubicada la estación, es 14° , dando un ángulo $\beta = 29^\circ$. Esto para la optimización de la luz solar, durante un periodo de 12 horas.

Figura 133. **Resultado final de estación meteorológica impresa en 3D y acople de sensor de calidad de aire**



Fuente: elaboración propia. INSIVUMEH.

Figura 134. **Entrega estación meteorológica a personal de INSIVUMEH**



Fuente: elaboración propia. INSIVUMEH.

En la figura 134 se realizó la entrega de la estación elaborada en el EPS, al Jefe del Departamento de Investigación y Servicios Meteorológicos y Jefe de Investigación de Investigación de Servicios Hídricos, en las instalaciones de INSIVUMEH central.

5.5. Manual de usuario

Este manual es una breve descripción de los componentes de instalación meteorológica, enfocados principalmente para las estaciones tipo automática. Una de las características de estas estaciones es su manera de operar, ya que trabajan de forma independiente y se alimentan de energía a través de un set de baterías que son recargadas por paneles solares, de esta manera envía información de forma continua en intervalos de tiempo de 1 minuto y almacenan la información en un servidor en la nube y/o local, en donde los datos pueden ser consultados en cualquier momento.

5.5.1. *Instalación de una estación meteorológica*

El usuario final para las estaciones meteorológicas debe considerar, según su requerimiento de adquisición de datos, los siguientes aspectos; con la finalidad de mejorar la calidad del dato registrado mediante los distintos sensores.

5.5.1.1. *Ubicación de la estación*

La ubicación es importante para instalar una estación meteorológica, porque debe cumplir con ciertos requerimientos que garanticen la calidad, la recolección y la transmisión de los datos. Para ello debe cumplir con las siguientes características:

- Que represente el clima del lugar.
- De fácil acceso y ubicación segura en cuento a robo.
- Se deben respetar las dimensiones de área libre de obstáculos alrededor de la estación.
 - Condición ideal: en un diámetro de 100 metros, la estación debe estar libre de árboles y obstáculos mayores de 4 metros.
 - Condición de operación mínima. En un diámetro de 30 metros, la estación debe estar completamente despejada de obstáculos.
- El lugar debe estar sobre o cerca de planicies o algún lugar con poca pendiente para que no esté expuesto a la erosión.
- El lugar debe estar alejado de fábricas, carreteras asfaltadas o cualquier otro lugar que pueda ser foco calorífico.
- El lugar debe de estar alejado da autopistas, ferrocarriles o maquinarias que pueden ocasionar vibraciones.
- El suelo debe estar cubierto de césped o vegetación natural de la región que no tenga más de 5 centímetros.

- Circular el área circundante con malla metálica.
- Tener buena señal GPRS para transmitir los datos al servidor.
- Permitir el acceso al personal encargado del mantenimiento preventivo mensual.

5.5.1.2. Infraestructura de la estación

- Para la construcción de la estación se debe cumplir con lo siguiente:
- Área de 10 x 10 metros.
- Circulada con malla metálica y alambre espigado o *razor* en la parte superior de la malla.
- La base de la malla metálica debe estar fundida a una o dos hileras de block en todo el perímetro.
- Orientada al norte.

5.5.1.3. Sistema de protección contra descargas electro-atmosféricas

Utilizar el sistema de protección tipo Rehilete, el cual funciona desviando las descargas eléctricas para que no aterricen en la estación meteorológica.

- Para la instalación se debe seguir los siguientes pasos:
- Instalar el pararrayos en la base de este a una altura de 3m sobre la punta del mástil.
- Instalar el cable calibre AWG 1/0 que une la punta del pararrayos con las cajas de disipación y fijar el mástil con flejes a cada metro.
- El cable debe seguir una trayectoria lo más recto posible, las curvas que no se puedan evitar deberán tener un radio igual o mayor a 0,20m a

manera de evitar alguna fuga de electricidad al momento de alguna descarga.

5.6. Manual de mantenimiento

Se debe realizar mantenimientos preventivos mensualmente, esto con el fin de evitar averías ocasionadas por el uso, desgaste o paso del tiempo.

5.6.1. *Hardware*

Para realizar el mantenimiento preventivo debe seguir los siguientes pasos:

- Desde la planificación realizar un chequeo de la transmisión y validación de datos.
- Revisión visual en *display* de regulador de carga del panel solar.
- Verificación de valor de batería con multímetro.
- Apagar la estación meteorológica.
 - Apagar *Raspberry Pi* desde SSH desde oficinas centrales, con el comando “*sudo shutdown -h now*” o llevar pantalla, teclado y mouse portátil para trabajar en campo.
 - Desconecte todas las terminales del regulador de carga del panel solar.
 - Retire el conector de sensores acoplados a los GPIO de *Raspberry Pi*.
- Realizar limpieza con limpia contactos y aire comprimido en todas las terminales.
- Verificación de sensores.
- Realizar limpieza a cada sensor, principalmente a sensores de precipitación e intensidad lumínica.

- Conecte nuevamente las terminales.
 - Conecte el adaptador de los GPIO a *Raspberry Pi*.
 - Conecte nuevamente las terminales del regulador de carga del panel solar. La *Raspberry Pi* se iniciará automáticamente.
- Verificar la transmisión de datos a servidor.

5.6.2. Software

Para realizar el mantenimiento preventivo debe realizar los siguientes pasos:

- Parar procesos de *script*, usando el comando “*sudo crontab -e*”.
 - Comentar las líneas de ejecución de proceso. Ver figura 134.
 - Guardar cambios y aceptar.
- Realizar actualización de sistema, ingresar a otra terminal y ejecutar los siguientes comandos:
 - *Sudo apt-get update*.
 - *Sudo apt-get upgrade*.
- Iniciar procesos de *script*, usando el comando “*sudo crontab -e*”
 - Descomentar las líneas de ejecución de proceso. Ver figura 134.
 - Guardar cambios y aceptar.

Nota: se debe realizar la actualización únicamente de paquetes esenciales de *Raspberry Pi*, no se incluyen actualizaciones de librerías para sensores, ya que pueden necesitar cambios en la programación implementada.

Se recomienda realizar reinicios periódicos al sistema en lapsos de seis horas, esto lo puede realizar eliminando el símbolo # de la última línea en las líneas de ejecución. Ver figura 135. Si existe el riesgo de pérdida de datos

significativos durante la temporada lluviosa en la región, puede realizar el reinicio del equipo en periodos mensuales.

Figura 135. Fichero de *script*

```

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# 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

## Sensor BMP280 Presion
*/1 * * * * python /home/pi/software/wx_stn/BMP280_pressure_temperature/bmp280_measure_2file_2web.py >> /tmp/bmp280_log_2x41

## Sensor HTU21D Humedad
*/1 * * * * python /home/pi/software/wx_stn/HTU21D_temperature_humidity/htu21d_read_2file_2web.py >> /tmp/htu21d_log_2x41

## Sensor MCP9808 Temperatura
*/1 * * * * python /home/pi/software/wx_stn/MCP9808_temperature/mcp9808_measure_2file_2web.py >> /tmp/mcp9808_log_2x41

## Sensor SI1145 Intensidad Luminica
*/1 * * * * python /home/pi/software/wx_stn/SI1145_UV_light/si1145_read_2file_2web.py >> /tmp/si1145_log_2x41

## Precipitation Tipping Bucket
@reboot python /home/pi/software/wx_stn/rain/tipping_bucket_2file_2web.py >> /tmp/rain_log_2x41

## Direccion del Viento
@reboot python /home/pi/software/wx_stn/winddir/winddir_minAvg_2file_2web.py >> /tmp/winddir_log_2x41

## Velocidad del Viento
@reboot python /home/pi/software/wx_stn/windspeed/windspd_measure_2file_2web.py >> /tmp/windspd_log_2x41

## Rebooteo de i2c necesario para cada reboot
@reboot ln -s /dev/i2c-1 /dev/i2c-0

## Opción de reboot cada 6 horas
0 5 2,8,14,20 * * * /sbin/reboot
    
```

Fuente: elaboración propia, empleando captura de pantalla.

Tabla XXV. Propuesta de cronograma para mantenimiento

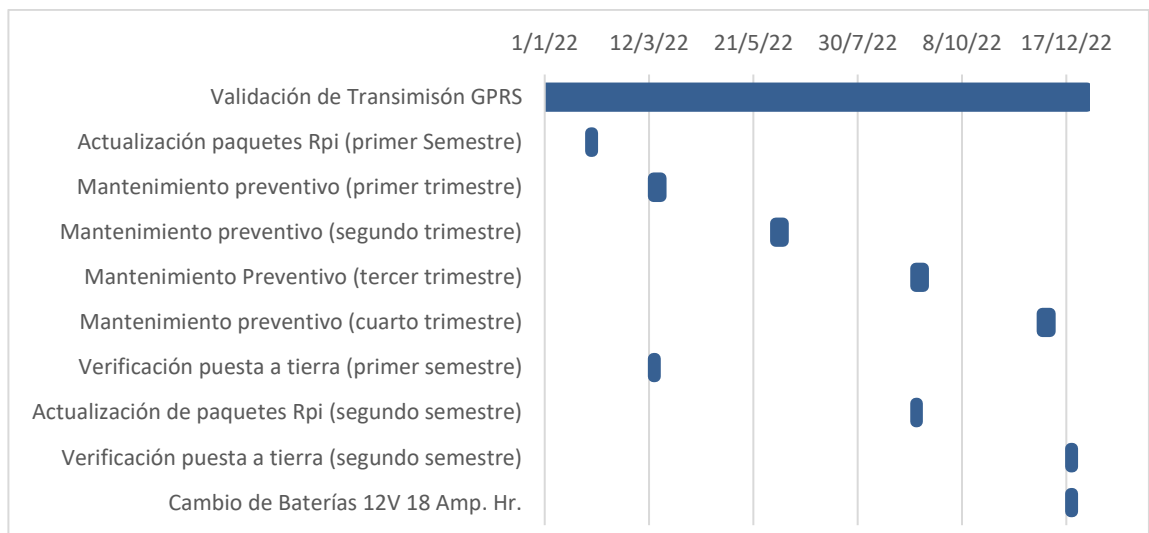
Nombre de Actividad	Fecha de inicio	Cantidad de días	Fecha finalización
Validación de Transmisión GPRS	1/01/2022	364	31/12/2022
Actualización paquetes Rpi (primer Semestre)	1/02/2022	1	2/02/2022
Mantenimiento preventivo (primer trimestre)	15/03/2022	5	20/03/2022
Mantenimiento preventivo (segundo trimestre)	5/06/2022	5	10/06/2022
Mantenimiento Preventivo (tercer trimestre)	7/09/2022	5	12/09/2022
Mantenimiento preventivo (cuarto trimestre)	1/12/2022	5	6/12/2022

Continuación de la tabla XXV.

Verificación puesta a tierra (primer semestre)	15/03/2022	1	16/03/2022
Actualización de paquetes Rpi (segundo semestre)	7/09/2022	1	8/09/2022
Verificación puesta a tierra (segundo semestre)	20/12/2022	1	21/12/2022
Cambio de Baterías 12V 18 Amp. Hr.	20/12/2022	1	21/12/2022

Fuente: elaboración propia, empleando Microsoft Word 365.

Figura 136. **Programación de mantenimiento propuesto**



Fuente: elaboración propia, empleando Microsoft Excel 365.

Tabla XXVI. **Formato de programación de mantenimiento**

Nombre de Actividad	Fecha de inicio	Cantidad de días	Fecha de finalización
Validación de Transmisión GPRS			
Actualización paquetes Rpi (primer Semestre)			
Mantenimiento preventivo (primer trimestre)			
Mantenimiento preventivo (segundo trimestre)			
Mantenimiento Preventivo (tercer trimestre)			
Mantenimiento preventivo (cuarto trimestre)			
Verificación puesta a tierra (primer semestre)			
Actualización de paquetes Rpi (segundo semestre)			
Verificación puesta a tierra (segundo semestre)			
Cambio de Baterías 12V 18 Amp. Hr.			

Fuente: elaboración propia, empleando Microsoft Excel 365.

CONCLUSIONES

1. Se presenta el diseño y la implementación de una estación meteorológica impresa en 3D, con la utilización de un *Raspberry Pi 3* modelo b como procesador de señales adquiridas por los distintos sensores.
2. Se incorpora la utilización de un módulo GPRS Sim900 para la transmisión de datos a un servidor de INSIVUMEH.
3. Se desarrolla código de programación para cada variable registrada de forma individual, haciendo uso de manejo de archivos para guardar los datos en la memoria disponible de la *Raspberry Pi*.
4. Se incorpora el uso de Access Point Name (APN) propia de INSIVUMEH para obtener una comunicación directa desde la estación meteorológica hacia un servidor de la institución.
5. Se genera la transferencia de datos de manera automática de los parámetros meteorológicos registrados en una estación meteorológica de bajo costo, impreso en 3D, hacia una base de datos alojada en un servidor de INSIVUMEH.

RECOMENDACIONES

1. Implementar a la red de estaciones meteorológicas de INSIVUMEH estaciones de bajo costo, con la utilización de un computador de tamaño reducido *Raspberry Pi* como uso de registrador de datos, por su versatilidad.
2. Mantener una transmisión de datos constantes hacia el servidor, se puede considerar la implementación de envío de datos satelital, por fallas que pueda presentar el servicio proporcionado por el proveedor.
3. Considerar, al momento de implementar los sensores, se permite realizar un concatenado de los archivos para generar un solo código y enviar los datos con una sola línea de comando.
4. Considerar, en lugares donde no se posea cobertura del proveedor de servicio del SIM, se puede incorporar el uso de un SIM con cobertura internacional y realizar la comunicación mediante el uso de una red privada virtual, conocida por sus siglas como VPN.
5. Considerar mantener el respaldo de información recolectada de las distintas variables registradas en la memoria interna microSD de la *Raspberry Pi*, por fallas de transmisión de datos.

BIBLIOGRAFÍA

1. MALDONADO, Isaac; RUIZ, Rubén; FUENTES, Marcel. *Manual de mantenimiento de estaciones meteorológicas automáticas*. Chile: INIA, 2010. 71 p.
2. Microchip. *Datasheet: BMP280*. [en línea]. <<https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>>. [Consulta: 12 de marzo de 2021].
3. Microchip. *Datasheet: HTU21D*. [en línea]. <https://cdn-shop.adafruit.com/datasheets/1899_HTU21D.pdf>. [Consulta: 12 de marzo de 2021].
4. Microchip. *Datasheet: Littelfuse 55300-00-002-A*. [en línea]. <https://m.Littelfuse.com/~media/electronics/datasheets/hall_effect_sensors/Littelfuse_hall_effect_sensors_55300_datasheet.pdf.pdf>. [Consulta: 12 de marzo de 2021].
5. Microchip. *Datasheet: MCP9808*. [en línea]. <<https://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf>>. [Consulta: 12 de marzo de 2021].
6. Microchip. *Datasheet: SI1145*. [en línea]. <<https://cdn-shop.adafruit.com/datasheets/Si1145-46-47.pdf>>. [Consulta: 12 de marzo de 2021].

7. Microchip. *Datasheet: SS451A.* [en línea]. <https://sensing.honeywell.com/honeywell-sensing-omnipolar-digital-hall-effect-sensor-ic-datasheet-ss351at_ss451a_ss551at_005917-4-en.pdf>. [Consulta: 12 de marzo de 2021].
8. NAVARRETE JUNGAL, Franklin Fernando. *Diseño e implementación de un prototipo de sistema de recolección de datos sísmicos basados en tecnología GPRS y la plataforma Raspberry Pi 3.* Trabajo de graduación de Ing. en Electrónica y Telecomunicaciones. Facultad de Ingeniería Eléctrica y Electrónica. Escuela Politécnica Nacional. Ecuador. 2019. 96 p.
9. RELANCIO ROSEL, Alberto. *Creación y administración de bases de datos.* [en línea]. <https://citarea.cita-aragon.es/citarea/bitstream/10532/4023/1/2018_048.pdf>. [Consulta: 12 de marzo de 2021].
10. THEISEN, Adam; UNGAR, Max; SHERIDAN, Bryan; ILLSTON, Bradley. *More science with less: evaluation of a 3D-printed weather station.* [en línea]. < <https://amt.copernicus.org/articles/13/4699/2020/amt-13-4699-2020.pdf>>. [Consulta: 12 de marzo de 2021].
11. TOBAJAS GARCÍA, Alberto. *Diseño e implementación de una estación meteorológica con Raspberry Pi.* Trabajo de graduación de Maestría en Ingeniería de Telecomunicaciones. Facultad de Ingeniería. Universidad de Abierta de Cataluña. España. 2016. 174 p.

APÉNDICES

A continuación, se muestra el código para cada uno de los sensores integrados.

Apéndice 1. Programa sensor dirección de viento desarrollado en *Python3*

```
1  #!/usr/bin/python3
2  # Código para control del sensor de dirección de viento. Escribe observaciones en disco
3  # calcula una media geométrica de 1 minuto.
4
5  #Presentacion BD
6  db = MySQLdb.connect("172.21.9.52","root","root","datosinsivumeh")
7  cursor=db.cursor()
8
9  #Import librerias a utilizar
10 import spidev, time, os, glob, csv, datetime
11 import MySQLdb
12 import mysql.connector
13 import pymysql
14 spi = spidev.SpiDev()
15 spi.open(0,0)
16
17 import RPi.GPIO as GPIO
18 GPIO.setwarnings(False)
19 GPIO.setmode(GPIO.BOARD)
20 GPIO.setup(13, GPIO.OUT)
21
22 import requests
23
24 # Sleep time, es necesario tener en cuenta el procesamiento de ~0.03 segundos
25 SLEEP_TIME = 0.97
26
27 # Channel 0 es para direccion de viento
28 def analog_read(channel):
29     if ((channel > 1) or (channel < 0)):
30         return -1
31
32     r = spi.xfer2([1, (2 + channel) << 6, 0])
33     adc_out = ((r[1]&31) << 6) + (r[2]>>2)
34     return adc_out
35
36 # Toma de fecha actual
37 now_file = datetime.datetime.now()
38
39 # Conteo de viento inicial
40 wnd_cnt = 1
41 time_cnt = 1
42
```

Continuación del apéndice 1.

```
43 # Lectura de viento desde el sensor
44 while True:
45     # Sensor de viento esta en channel 0
46     ch = 0
47
48     # lectura del canal
49     reading = analog_read(ch)
50
51     # Entrada min/max de ADC tomada de calibracion
52     ADC_min = 103
53     ADC_max = 920
54     ADC_range = ADC_max - ADC_min
55
56     # Conversion de grados desde el norte
57     if (reading < ADC_min):
58         wnddir = 0
59     elif (reading > ADC_max):
60         wnddir = 360
61     else:
62         # Toma de ADC a cero relativo
63         ADC_rel = reading - ADC_min
64         wnddir = ADC_rel*360.0/ADC_range
65
66     # Registra el primer angulo
67     if(time_cnt == 1):
68         wnddir_sum = wnddir
69         wnddir_prev = wnddir
70         time_cnt = time_cnt + 1
71     elif (time_cnt < 60):
72         delta_wnd = wnddir - wnddir_prev
73         # Asigna el siguiente valor
74         if( delta_wnd < -180.0):
75             wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd + 360.0)
76             wnddir_prev = wnddir_prev + delta_wnd + 360.0
77             wnd_cnt = wnd_cnt + 1
78         elif (abs(delta_wnd) < 180.0):
79             wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd)
80             wnddir_prev = wnddir_prev + delta_wnd
81             wnd_cnt = wnd_cnt + 1
82         elif (delta_wnd > 180):
83             wnddir_sum = wnddir_sum + (wnddir_prev + delta_wnd - 360.0)
84             wnddir_prev = wnddir_prev + delta_wnd - 360.0
85             wnd_cnt = wnd_cnt + 1
```


Apéndice 2. Programa sensor velocidad de viento desarrollado en *Python3*

```
1  #!/usr/bin/python3
2  # Codigo para controlar el sensor de velocidad del viento. Escribe la observacion en el disco
3
4  #Presentacion BD
5  db = MySQLdb.connect("172.21.9.52","root","root","datosinsivumeh")
6  cursor=db.cursor()
7
8  # importar librerias
9  import RPi.GPIO as GPIO
10 import time, datetime
11 import MySQLdb
12 import mysql.connector
13 import pymysql
14 import requests
15
16 # Variable de contador del sensor
17 COUNT = 1
18
19 # Numero de sensor en anemometro
20 SENSOR_NUM = 2
21
22 # Tiempo de muestra en segundos
23 SAMPLE = 58.5
24
25 # Sleep time
26 SLEEP_TIME = 60
27
28 # Factor de calibración de la velocidad del viento
29 CAL_Factor = 2.64 #(3.14/1.19) - Factor original de calibracion
30
31 # Factor de compensación para diferentes materiales (PLA)
32 #OFFSET = 1.43
33 # Need to have an offset of 0.0
34 OFFSET = 0.0
35
36 SCALE = CAL_Factor*(2*3.14156*0.079)/(SENSOR_NUM*SAMPLE) # Velocidad de viento en m/s
37
38 # Identificando el pin GPIO para el sensor de viento
39 PIN = 22
40
41 GPIO.setmode(GPIO.BCM)
42 GPIO.setup(PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```


Continuación del apéndice 2.

```
89     # Escribir datos en la pantalla
90     line = "%02d %02d %4d %02d %02d %02d %4.2f" % (now.month, now.day, now.year, now.hour, now.minute, now.second, wind_spd)
91
92     # Imprimir en la pantalla
93     print(line)
94
95     #flush - Escribe en archivo en tiempo real
96     file.flush()
97     file2.flush()
98
99     # Cerrar el archivo para la proxima iteracion
100    file.close()
101    file2.close()
102
103    wind = 0
104
105    # Sleep durante 60 segundos, incluido el segundo actual
106    time.sleep(SLEEP_TIME)
107
108    # Cerrar el archivo de registro
109    file.close()
110    file2.close()
111    GPIO.cleanup()
```

Fuente: elaboración propia, empleando *Python3*.

Apéndice 3. Programa sensor de HTU21D desarrollado en *Python2*

```
1  #!/usr/bin/python
2  #Codigo para contor de sensor HTU21D temp y humedad. Escribir observaciones en disco
3  #Presentacion BD
4  db = MySQLdb.connect("172.21.9.52","root","root","datosinsivumeh")
5  cursor=db.cursor()
6  |
7  # import librerias necesarias
8  import sys, struct, array, time, io, os, fcntl, datetime
9  import MySQLdb
10 import mysql.connector
11 import pymysql
12
13 import requests
14
15 I2C_SLAVE=0x0703
16 HTU21D_ADDR = 0x40
17 CMD_READ_TEMP_HOLD = "\xE3"
18 CMD_READ_HUM_HOLD = "\xE5"
19 CMD_READ_TEMP_NOHOLD = "\xF3"
20 CMD_READ_HUM_NOHOLD = "\xF5"
21 CMD_WRITE_USER_REG = "\xE6"
22 CMD_READ_USER_REG = "\xE7"
23 CMD_SOFT_RESET= "\xFE"
24
25 class i2c(object):
26     def __init__(self, device, bus):
27
28         self.fr = io.open("/dev/i2c-"+str(bus), "rb", buffering=0)
29         self.fw = io.open("/dev/i2c-"+str(bus), "wb", buffering=0)
30
31         # Se setea la direccion del dispositivo
32         fcntl.ioctl(self.fr, I2C_SLAVE, device)
33         fcntl.ioctl(self.fw, I2C_SLAVE, device)
34
35     def write(self, bytes):
36         self.fw.write(bytes)
37
38     def read(self, bytes):
39         return self.fr.read(bytes)
40
41     def close(self):
42         self.fw.close()
43         self.fr.close()
44
45 class HTU21D(object):
46     def __init__(self):
47         self.dev = i2c(HTU21D_ADDR, 1) #HTU21D 0x40, bus 1
48         self.dev.write(CMD_SOFT_RESET) #soft reset
49         time.sleep(.1)
50
51     def ctemp(self, sensorTemp):
52         tSensorTemp = sensorTemp / 65536.0
53         return -46.85 + (175.72 * tSensorTemp)
54
55     def chumid(self, sensorHumid):
56         tSensorHumid = sensorHumid / 65536.0
57         return -6.0 + (125.0 * tSensorHumid)
58
59     def crc8check(self, value):
60         # Importado desde Sparkfun Arduino HTU21D Library: https://github.com/sparkfun/HTU21D_Breakout
61         remainder = ( ( value[0] << 8 ) + value[1] ) << 8
62         remainder |= value[2]
63
64         # Polinomio = 0x0131 = x^8 + x^5 + x^4 + 1
65         # divisor = 0x988000 es el 0x0131 polinomio desplazado al extremo izquierdo a tres bytes
66         divsor = 0x988000
67
68         for i in range(0, 16):
69             if( remainder & 1 << (23 - i) ):
70                 remainder ^= divsor
71                 divsor = divsor >> 1
```

Continuación del apéndice 3.

```
72
73     if remainder == 0:
74         return True
75     else:
76         return False
77
78     def read_temperature(self):
79         self.dev.write(CMD_READ_TEMP_NOHOLD) #medicion de temperatura
80         time.sleep(.1)
81
82         data = self.dev.read(3)
83         buf = array.array('B', data)
84
85         if self.crc8check(buf):
86             temp = (buf[0] << 8 | buf [1]) & 0xFFFC
87             return self.ctemp(temp)
88         else:
89             return -255
90
91     def read_humidity(self):
92         self.dev.write(CMD_READ_HUM_NOHOLD) #medicion de humedad
93         time.sleep(.1)
94
95         data = self.dev.read(3)
96         buf = array.array('B', data)
97
98         if self.crc8check(buf):
99             humid = (buf[0] << 8 | buf [1]) & 0xFFFC
100            return self.chumid(humid)
101        else:
102            return -255
103
104 if __name__ == "__main__":
105     obj = HTU21D()
106     tempC = obj.read_temperature()
107     tempF = obj.read_temperature()*1.8+32
108     rh = obj.read_humidity()
109     print "HTU21D Sensor"
110     print "Temperature:", '{:.2f}'.format(obj.read_temperature()), "degC"
111     print "Temperature:", '{:.2f}'.format((obj.read_temperature()*1.8+32)), "degF"
112     print "Humidity:", '{:.0f}'.format(obj.read_humidity()), "%"
113
114     # Obener la fecha actual
115     now = datetime.datetime.now()
116
117     # Agregar la seccion para escribir el archivo
118     # Abrir y editar el archivo
119     # Nombre de archivo basado en fecha
120     filename = '/home/pi/data/wx_stn/HTU21D/htu21d_%4d%02d%02d.dat' % (now.year, now.month, now.day)
121     filename2 = '/home/pi/Desktop/Concatenado/Humedad.txt'
122
123     # Abrir archivo
124     file = open(filename, 'a')
125     file2 = open(filename2, 'a')
126
```


Apéndice 5. Programa sensor BMP280 desarrollado en Python2

```
1  #!/usr/bin/python
2  #Codigo para controlar el sensor BMP280 I2C. Escribe observaciones en el disco
3  #Presentacion BD
4  db = MySQLdb.connect("172.21.9.52","root","root","datosinsivumeh")
5  cursor=db.cursor()
6
7  # Importar librerias
8  import Adafruit_BMP.BMP280 as BMP280
9
10 import sys, struct, array, time, io, os, fcntl, datetime
11 import MySQLdb
12 import mysql.connector
13 import pymysql
14 import requests
15
16 # Se inicializa el sensor BMP085 y se usa el modo estandar (default value)
17 #bmp = BMP085(0x77)
18 bmp = BMP280.BMP280()
19
20 temperature = bmp.read_temperature()
21
22 # Leer el nivel actual de presión barométrica actual
23 pressure = bmp.read_pressure()
24
25 # Para calcular la altitud en función de una presión media estimada del nivel del mar
26 # (1013.25 hPa) llamar a la funcion de la siguiente manera, pero esto no será muy preciso
27 altitude = bmp.read_altitude()
28
29 # Para especificar una altitud más precisa, ingrese el nivel de presión al nivel del mar medio correcto
30 # Por ejemplo, si el nivel de presión actual es 1023.50 hPa,
31 # ingrese 102350 ya que incluimos dos decimales en valor entero
32 # altitude = bmp.read_altitude(102350)
33
34 slp = (pressure/100.0)*pow(1-(0.0065*altitude)/(temperature+0.0065*altitude+273.15),-5.257)
35
36 print "BMP180 Sensor"
37 print "Temperature: ", '{:.2f}'.format(temperature), "deg C"
38 print "Temperature: ", '{:.2f}'.format((temperature*1.8+32)), "deg F"
39 print "Station Pressure: ", pressure/100.0, " mb"
40 print "Sea Level Pressure: ", '{:.2f}'.format(slp), " mb"
41 print "Sea Level Pressure: ", '{:.2f}'.format(slp/33.86389), " inHg"
42 print "Altitude:   %.2f m" % altitude
43
44 # Obtener fecha actual
45 now = datetime.datetime.now()
46
47 # Abrir archivo y agregar
48 # Nombre de archivo basado en fecha actual
49 filename = '/home/pi/data/wx_stn/BMP280/bmp280_%d%02d%02d.dat' % (now.year, now.month, now.day)
50 filename2 = '/home/pi/Desktop/Concatenado/Presion.txt'
51
52 #Abrir archivo
53 file = open(filename, 'a')
54 file2 = open(filename2, 'a')
55
56 try:
57     # Variables medidas o derivadas
58     tempC = temperature
59     tempF = temperature*1.8+32
60     slp = (pressure/100.0)*pow(1-(0.0065*altitude)/(tempC+0.0065*altitude+273.15),-5.257)
61     station_pres = pressure/100.0
62     slp_hg = slp/33.86389
63
```


Continuación del apéndice 5.

```
64 # Escribir archivo
65 file.write('%02d %02d %4d %02d %02d %0.2f %0.2f %0.2f %7.2f %5.2f %7.2f\n' % (now.month, now.day, now.year, now.hour, now.minute, tempc, tempF, station_pres, slp, slp_hg, altitude))
66 file2.write('%0.2f %7.2f %5.2f %7.2f\n' % (station_pres, slp, slp_hg, altitude))
67
68 # Escribir en base de datos
69 sql = "INSERT INTO estacion_ed(,,,,, ,Presion ,PhivMar_hPa ,PhivMar_inHg ,Altitud ,,,,,) VALUES(,,,,, ,%0.2f ,%7.2f ,%5.2f ,%7.2f ,,,,,)"
70 cursor.execute(sql)
71 db.commit()
72
73 finally:
74 if file is not None:
75     file.close()
76     file2.close()
```

Fuente: elaboración propia, empleando *Python2*.

Apéndice 6. Programa para variable precipitación desarrollado en *Python3*

```
1 #!/usr/bin/python3
2 #Codigo para controlar el tipping bucket pluviometro. Escribir observaciones en disco
3 |
4 #Presentacion BD
5 db = MySQLdb.connect("172.21.9.52","root","root","datosinsivumeh")
6 cursor=db.cursor()
7
8 #importar librerias necesarias
9 import RPi.GPIO as GPIO
10 import time, datetime
11 import MySQLdb
12 import mysql.connector
13 import pymysql
14 import requests
15
16 # Acumulado de lluvia en mm por punta de cuchara
17 CALIBRATION = 0.2
18
19 # Identificando GPIO para sensor de lluvia
20 PIN = 23
21
22 # Sleep time entre mediciones
23 SLEEP_TIME = 58.5
24
25 GPIO.setmode(GPIO.BCM)
26 GPIO.setup(PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
27
28 # Variable para realizar un seguimiento de la cantidad de lluvia
29 rain = 0
30
31 # Funcion de devolucion de llamada para cada punta de cubo
32 def cb(channel):
33     global rain
34     rain = rain + CALIBRATION
35
36 # Registrar la devolucion de llamada para interrupciones de pin
37 GPIO.add_event_detect(PIN, GPIO.FALLING, callback=cb, bouncetime=300)
38
39 # Obtener fecha actual
40 now_file = datetime.datetime.now()
41
```




Apéndice 8. Fichero rnet para configuración GPRS en RPi

```
GNU nano 3.2                                rnet                                Modificado
#indicamos la apn a conectarnos
connect "/usr/bin/chat -v -f /etc/chatscripts/gprs -T ebis.tigo.gt"
#indicamos el puerto de comunicacion
/dev/ttyUSB0
#definimos el baudrate de funcionamiento
115200
#intenta obtener la dirección IP
usepeerdns
#Utiliza esta conexión como ruta predeterminada
defaultroute
#Hace que PPP vuelva a marcar cuando se pierda la conexión
persist
#no le pide al control remoto que se autentique
noauth
#Sin control de flujo de hardware en el enlace serial con modem
nocrtscts
#sin líneas de control de modem
local

^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar txt^J Justificar^C Posición
^X Salir      ^R Leer fich.^_ Reemplazar ^U Pegar txt  ^T Ortografía^_ Ir a línea
```

Fuente: elaboración propia, empleando *Python2*.

Apéndice 9. Hoja para visita de campo

		INSTITUTO NACIONAL DE SISMOLOGÍA, VULCANOLOGÍA, METEOROLOGÍA E HIDROLOGÍA - INSIVUMET - UNIVERSIDAD DE SAN CARLOS DE GUATEMALA - USAC -			
PROYECTO DE EPS Estación Bajo Costo - Meteorológica / Calidad del Aire CONTROL DE VISITA/ ESTACION					
Fecha de Visita		PROYECTO		PERSONAL ASIGNADO	
TIPO DE VISITA		SUPERVISOR:		Carlos Montenegro	
SOPORTE TÉCNICO:		TECNICO:			
DATOS ESTACION Meteorológica Bajo Costo (3D)					
NOMBRE DE LA ESTACION		MUNICIPIO:		TIPO DE ESTACION	
DEPARTAMENTO: GUATEMALA		GUATEMALA		AUTOMÁTICA	
TRANSMISION SATELITAL		TRANSMISION GPRS		LONGITUD: -90.532621	
TRANSMISION: NO		SI		ALTIUD: 1500.34 m.s.n.m	
ID GOES: F2360778		TRANSMISION: NO		COMUNICACION	
ORIENTACION:		No. TELEFONO: 40403789		SERVICIO: VPN	
		IP ASIGNADO: 10.251.112.16		TRANSMISION SI / NO	
		BANDA:		IP	
DETALLE DE EQUIPO AUTOMATICO					
No.	Cantidad	Descripción	ESTADO	Trabajo Realizado	OBSERVACIONES
1	1	Raspberry Meteorológica	Bueno	MANTENIMIENTO PREVENTIVO	
2	1	Raspberry Calidad del Aire	Bueno	MANTENIMIENTO PREVENTIVO	
3	1	Anemómetro	Bueno	MANTENIMIENTO CORRECTIVO	
4	1	Veleta	Bueno		
5	1	Temperatura	Bueno		
6	1	Humedad	Bueno		
7	1	Presión	Bueno		
8	1	Pluviómetro	Bueno		
9	1	Sensor luminico (IR y UV)	Bueno		
10	1	Sensor NOVA SDS011	Bueno		
11	1	Luz Estroboscópica	Bueno		
12	2	Panel solar 20 W	Bueno		
13	2	Bat. 12 V	Bueno	MANTENIMIENTO CORRECTIVO	
14	1	GPRS	Bueno		
15	1	GOES	Bueno		

Fuente: elaboración propia, empleando Microsoft Excel 365.