

**Maestría en Ingeniería Electrónica**

**DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO DE UN SISTEMA DE  
ADQUISICIÓN, TRANSMISIÓN Y VISUALIZACIÓN DE DATOS  
HIDROMETEOROLÓGICOS.**

**Manuel Felipe Rodríguez Pérez**



**Bogotá, D.C., 13 de Febrero de 2018**

**DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO DE UN SISTEMA DE  
ADQUISICIÓN, TRANSMISIÓN Y VISUALIZACIÓN DE DATOS  
HIDROMETEOROLÓGICOS.**

**Tesis para optar al título de magíster en Ingeniería Electrónica,  
con énfasis en automatización industrial**

**Ing. Javier Soto Vargas PhD.**

**Director del Trabajo**

**Jurado: Juan Manuel López López**

**Jurado: Jesús Fidernan Barrera Cobos**



**Bogotá, D.C., 13 de Febrero de 2018**

Nota de Aceptación:

La tesis de maestría titulada “DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO DE UN SISTEMA DE ADQUISICIÓN, TRANSMISIÓN Y VISUALIZACIÓN DE DATOS HIDROMETEOROLÓGICOS, presentada por Manuel Felipe Rodríguez Pérez, cumple con los requisitos establecidos para optar al título de Magíster en Ingeniería Electrónica con énfasis en Automatización Industrial.

---

Jurado

---

Jurado

Bogotá, D.C, 13 de Febrero de 2018

*Dedico este trabajo a Dios, quien es el ser que me lo ha dado todo, a mis padres a quien debo mi formación como persona y profesional, a mis hermanos, sobrinos y demás familiares, a mi novia, compañeros de estudio, compañeros del IDEAM y a todas las personas que contribuyeron a la culminación de este trabajo de maestría.*

## AGRADECIMIENTOS

A mi familia:

**Melba Pérez, Alirio Rodríguez, Mauricio Rodríguez, Melba Rodríguez y Juan Camilo Rodríguez.**

Por su apoyo incondicional en el proceso de formación como ser humano y profesional.

A mi novia:

**María Andrea Pineda Fonseca**

Por su amor, cariño, comprensión y compañía diaria durante el desarrollo de todas las actividades de la Maestría.

A los compañeros y docentes:

Del programa de **Maestría en Ingeniería Electrónica, de la Escuela Colombiana de Ingeniería Julio Garavito**, por su respaldo y acompañamiento durante el proceso de formación académico. Especialmente al **Ing. Javier Soto Vargas** por sus aportes y asesoría como director de Proyecto de Grado.

A Directivos y Compañeros:

Del Instituto de Hidrología Meteorología y Estudios Ambientales (**IDEAM**) en cabeza de su director **Ing. Omar Franco Torres**, quienes alentaron y facilitaron elementos y herramientas necesarias para el desarrollo del proyecto, especialmente a todos los compañeros del **Grupo de Instrumentos y Metalmecánica**, y al **Grupo de Automatización** quienes con sus ideas, comentarios y conocimientos aportaron y apoyaron la realización del proyecto.

## RESUMEN

Este proyecto se presenta como trabajo de grado para optar al título de Magister en Ingeniería Electrónica, de la Escuela Colombiana de Ingeniería Julio Garavito, el cual se desarrolla con un enfoque hacia la comunidad colombiana interesada en registrar, procesar y analizar información en tiempo real de variables hidrometeorológicas de forma remota.

Se implementa el prototipo de un sistema de adquisición, transmisión y visualización de datos hidrometeorológicos, basado en un computador de placa reducida (**Raspberry Pi 2 Model B**), el cual permite interacción con variedad de dispositivos por medio de los pines entrada/salida de propósito general y diferentes interfaces de comunicación, con lo cual se implementan 8 entradas para **sensores análogos**, 2 entradas para **sensores digitales**, y 1 entrada para sensores seriales con protocolo **SDI-12**. Para la verificación del funcionamiento de las entradas análogas se utilizó un calibrador de procesos industriales en modo de generador de voltaje para realizar la comparación de voltaje ingresado contra voltaje leído en el prototipo a través del conversor análogo digital. Para la verificación de las entradas digitales por pulsos se utilizó un generador de pulsos manual con el cual se realizó la comparación de pulsos generados contra pulsos capturados por el prototipo en los pines GPIO de la Raspberry Pi. Finalmente para la verificación del funcionamiento de la entrada serial con protocolo SDI-12 se utiliza un sistema microcontrolado como analizador de protocolos (sniffer), con el fin de capturar la trama de datos generada entre el prototipo y un sensor SDI-12.

Adicionalmente el prototipo desarrollado integra una cámara fotográfica que permite la captura y almacenamiento de imágenes de acuerdo a una periodicidad preestablecida lo cual puede ser de gran aplicabilidad para registrar diferentes tipos de fenómenos atmosféricos, nubes, cambios ambientales como el deshielo o variables hidrológicas como niveles de ríos y embalses.

Dentro del computador y utilizando **Python3** se realiza la adquisición de la información proveniente de los sensores y se hace un procesamiento de los mismos, para luego almacenar los valores obtenidos en una base de datos local desarrollada con **MySql**.

En una pantalla táctil conectada a la Raspberry Pi y por medio de una interfaz gráfica desarrollada en **MonoDevelop** utilizando **C#**, se permite la configuración de parámetros de los sensores, gestión de la base de datos, análisis y visualización de imágenes y datos hidrometeorológicos provenientes de los sensores.

De acuerdo a una condición periódica definida por el usuario, la información es transmitida inalámbricamente vía internet utilizando la red celular **GPRS** en lugares donde se tenga cobertura de datos, para así, integrar toda la información de los sensores al sistema nacional de visualización utilizado en el Instituto de Hidrología Meteorología y Estudios Ambientales (**IDEAM**), conocido como **Hydras3**, el cual es la herramienta especializada utilizada por expertos para generar gráficas de comportamiento, tablas de datos y realizar análisis técnicos en el área de hidrología y meteorología.

# Índice General

<b>RESUMEN.....</b>	<b>6</b>
<b>INTRODUCCIÓN.....</b>	<b>17</b>
<b>1 OBJETIVOS.....</b>	<b>20</b>
1.1 OBJETIVO GENERAL .....	20
1.2 OBJETIVOS ESPECÍFICOS.....	20
<b>2 METODOLOGÍA DE DESARROLLO .....</b>	<b>21</b>
2.1 INGENIERÍA Y ANÁLISIS DEL SISTEMA.....	22
2.2 ANÁLISIS DE LOS REQUISITOS.....	22
2.3 DISEÑO .....	23
2.4 IMPLEMENTACIÓN (CODIFICACIÓN) .....	24
2.5 PRUEBA .....	25
2.6 MANTENIMIENTO.....	25
<b>3 MARCO REFERENCIAL.....</b>	<b>26</b>
3.1 OBSERVACIONES METEOROLÓGICAS A NIVEL MUNDIAL.....	26
3.2 OBSERVACIONES DE SUPERFICIE .....	28
3.2.1 <i>Observación en estaciones dotadas de personal (Convencionales)</i> .....	29
3.2.2 <i>Observación de datos de forma automática</i> .....	31
3.3 OBSERVACIONES METEOROLÓGICAS EN COLOMBIA .....	34
3.3.1 <i>Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM)</i> .....	34
3.3.2 <i>Otras entidades a nivel nacional</i> .....	36
3.4 BENEFICIOS DE LAS OBSERVACIONES METEOROLÓGICAS .....	36
3.4.1 <i>Prevención de desastres naturales</i> .....	37
3.4.2 <i>Generación de energía de fuentes renovables</i> .....	39
3.4.3 <i>Prevención de enfermedades</i> .....	40
3.4.4 <i>Producción alimentaria</i> .....	41
3.5 REVISIÓN DEL ESTADO DEL ARTE .....	41
3.6 SENSORES HIDROMETEOROLÓGICOS UTILIZADOS .....	45
3.6.1 <i>Sensor de temperatura y humedad del aire</i> .....	45
3.6.2 <i>Sensor de presión atmosférica</i> .....	46
3.6.3 <i>Sensor de precipitación</i> .....	47
3.6.4 <i>Sensor de dirección y velocidad del viento</i> .....	48
3.7 SOFTWARE DE MONITOREO REMOTO HYDRAS3 .....	50
3.7.1 <i>Versión de escritorio Hydras3</i> .....	51

3.7.2	<i>Versión WEB Hydras3</i> .....	54
<b>4</b>	<b>DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO</b> .....	<b>55</b>
4.1	PLATAFORMA DE DESARROLLO UTILIZADA (RASPERRY PI).....	55
4.2	CONFIGURACIÓN DEL RELOJ EN TIEMPO REAL.....	57
4.3	ADQUISICIÓN Y PROCESAMIENTO DE DATOS .....	59
4.4	MEDICIÓN DE VARIABLES HIDROMETEOROLÓGICAS .....	61
4.4.1	<i>Entradas análogas</i> .....	61
4.4.2	<i>Entradas digitales</i> .....	65
4.4.3	<i>Entrada serial SDI-12</i> .....	66
4.5	ALMACENAMIENTO DE INFORMACIÓN EN BASE DE DATOS.....	70
4.6	ADQUISICIÓN DE IMÁGENES .....	72
4.7	INTERFAZ GRÁFICA PARA CONFIGURACIÓN DE PARÁMETROS Y VISUALIZACIÓN DE DATOS .....	74
4.7.1	<i>Panel de configuración</i> .....	76
4.7.1.1	<i>Archivo de configuración XML</i> .....	79
4.7.2	<i>Panel de inicio</i> .....	79
4.7.3	<i>Panel de visualización gráfica de variables</i> .....	80
4.7.4	<i>Panel gestión base de datos</i> .....	81
4.7.5	<i>Panel de visualización de imágenes</i> .....	83
4.8	SISTEMA DE COMUNICACIÓN INALÁMBRICA .....	84
4.9	INTEGRACIÓN DE DATOS AL SISTEMA DE VISUALIZACIÓN METEOROLÓGICO HYDRAS3.....	85
4.9.1	<i>Archivo de datos ".MIS"</i> .....	86
4.9.2	<i>Plantilla de la estación XML</i> .....	88
4.10	SISTEMA DE ALIMENTACIÓN .....	88
<b>5</b>	<b>VERIFICACIÓN DE FUNCIONAMIENTO</b> .....	<b>90</b>
5.1	FUNCIONAMIENTO ENTRADAS ANÁLOGAS.....	90
5.2	FUNCIONAMIENTO ENTRADAS DIGITALES .....	94
5.3	FUNCIONAMIENTO ENTRADAS SERIAL SDI-12 .....	95
<b>6</b>	<b>RESULTADOS Y CONTRIBUCIÓN</b> .....	<b>98</b>
6.1	FABRICACIÓN DE CIRCUITOS IMPRESOS.....	98
6.2	RESULTADOS DE LA MEDICIÓN DE VARIABLES HIDROMETEOROLÓGICAS.....	100
6.3	INTEGRACIÓN DE DATOS AL SISTEMA DE VISUALIZACIÓN METEOROLÓGICO HYDRAS3.....	105
6.3.1	<i>Resultados de las pruebas de transmisión</i> .....	106
6.4	MONTAJE ESTRUCTURA FÍSICA.....	110
6.5	COSTOS DE FABRICACIÓN.....	113
<b>7</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>117</b>
<b>8</b>	<b>REFERENCIAS</b> .....	<b>121</b>

<b>ANEXO A. CONFIGURACIÓN RELOJ EN TIEMPO REAL RTC-DS1307 POR I2C .....</b>	<b>126</b>
<b>ANEXO B. CONFIGURACIÓN INTERFAZ SPI MÓDULO ADC MCP3008 .....</b>	<b>132</b>
<b>ANEXO C. INSTALACIÓN, CONFIGURACIÓN Y MANEJO INICIAL DE BASE DE DATOS MYSQL .....</b>	<b>139</b>
<b>ANEXO D. ESTUDIO DEL DIMENSIONAMIENTO DE UN SISTEMA DE ALIMENTACIÓN SOLAR FOTOVOLTAICO PARA EL PROTOTIPO DESARROLLADO. ....</b>	<b>144</b>
<b>ANEXO E. CODIGOS UTILIZADOS PARA EL DESARROLLO DE LA INTERFAZ GRÁFICA DE CONFIGURACION Y VISUALIZACION .....</b>	<b>150</b>
<b>ANEXO F. CODIGOS UTILIZADOS PARA EL FUNCIONAMIENTO DEL PROTOTIPO .....</b>	<b>151</b>

## Índice de Tablas

Tabla 1. Etapas y componentes del diseño del prototipo. ....	23
Tabla 2. Clasificación de estaciones hidrometeorológicas en Colombia.....	35
Tabla 3. Consolidado de atención a emergencias por eventos medioambientales 2014..	38
Tabla 4. Plataforma OTT NetDL1000 y NetDL500 .....	43
Tabla 5. Plataforma Sutron 8310 .....	43
Tabla 6. Plataforma SEBA UnilogCom.....	44
Tabla 7. Plataforma Siap+Micros DA15K.....	44
Tabla 8. Rangos de medición sensor de temperatura y humedad del aire .....	45
Tabla 9. Especificaciones sensor de dirección y velocidad del viento .....	49
Tabla 10. Comparación Raspberry Pi 2 Model B y Beaglebone Black Rev C .....	55
Tabla 11. Valores de sensibilidad de diferentes tipos de sensores. ....	62
Tabla 12. Convenciones de respuesta a comandos SDI-12.....	68
Tabla 13. Definición puntos de medición para calibración de multímetros digitales .....	90
Tabla 14. Puntos de medición para verificación de funcionamiento. ....	91
Tabla 15. Resumen de las funciones de fuente y medición – Fluke 725 .....	91
Tabla 16. Resultados de mediciones y cálculo de Incertidumbre .....	93
Tabla 17. Costos de fabricación unitaria del prototipo DCP. ....	115
Tabla 18. Comparación de especificaciones técnicas y costos de plataformas colectoras de datos con el prototipo DCP.....	116
Tabla 19. Diferentes tipos de sensores comerciales con la capacidad de ser integrados al prototipo desarrollado. ....	118
Tabla 20. Consumos de los elementos utilizados en el prototipo. ....	144
Tabla 21. Promedio horario de la radiación para la estación universidad nacional de Bogotá. ....	149

## Índice de Figuras

Figura 1. Clasificación de estaciones hidrometeorológicas en Colombia .....	18
Figura 2. Etapas del Modelo en Cascada .....	21
Figura 3. Diagrama de bloques diseño del prototipo .....	24
Figura 4. Sistema Global de Observación (GOS).....	27
Figura 5. Instrumentos convencionales.....	30
Figura 6. Instrumentos de registro autográfico.....	31
Figura 7. Sensores electrónicos.....	32
Figura 8. Plataformas colectoras de datos disponibles en el mercado.....	33
Figura 9. Tipos de telemetría en estaciones meteorológicas automáticas.....	34
Figura 10. Sensor de temperatura y humedad del aire .....	46
Figura 11. Sensor de presión atmosférica.....	46
Figura 12. Pluviómetro tipo balancín.....	47
Figura 13. Pluviómetro RIMCO 8500 .....	48
Figura 14. Sensor de dirección y velocidad del viento.....	48
Figura 15. Respuesta SDI-12 sensor de dirección y velocidad del viento .....	50
Figura 16. Interfaz de escritorio Hydras3 .....	51
Figura 17. Presentación gráfica y tabular – versión de escritorio software Hydras3.....	52
Figura 18. Superposición de múltiples gráficas en Hydras3.....	53
Figura 19. Configuración de límite de alarma Hydras3.....	53
Figura 20. Interfaz WEB Hydras3.....	54
Figura 21. Representación gráfica interfaz WEB Hydras3.....	54
Figura 22. Distribución de puertos e interfaces en la tarjeta Raspberry Pi .....	56
Figura 23. Captura de pantalla del sistema operativo Raspbian Jessie .....	57
Figura 24. Módulo RTC - DS1307.....	58

Figura 25. Conexión Raspberry Pi – Módulo DS1307 .....	58
Figura 26. Secuencia de inicio reloj de tiempo real .....	59
Figura 27. Estructura y componentes de software implementados en el prototipo. ....	60
Figura 28. Conversor análogo digital MCP3008.....	62
Figura 29. Conversor de niveles lógicos BSS138 .....	63
Figura 30. Conexión Raspberry Pi – BSS138 – MCP3008.....	63
Figura 31. Definición de la función analogmap para la conversión voltaje a variable .....	64
Figura 32. Conexión GPIO entradas digitales. ....	65
Figura 33. Funcionamiento de interrupción. ....	66
Figura 34. Ejemplo de interconexión del bus SDI-12.....	67
Figura 35. Conjunto básico de comandos/respuestas SDI-12.....	67
Figura 36. Diferentes tipos de conversores SDI-12.....	68
Figura 37. Ejemplo de conexión de sensores SDI-12 a Raspberry PI. ....	69
Figura 38. Conexión Raspberry Pi – Arduino Nano.....	70
Figura 39. Tablas implementadas en la base de datos .....	71
Figura 40. Módulo cámara Raspberry Pi.....	72
Figura 41. Pantalla y carcasa Raspberry Pi 7” .....	74
Figura 42. Captura de pantalla del sistema de desarrollo IDE MonoDevelop .....	75
Figura 43. Panel de configuración - Configuración inicial .....	76
Figura 44. Panel de configuración - Entradas análogas .....	77
Figura 45. Panel de configuración - Entradas digitales .....	78
Figura 46. Panel de configuración - Entrada serial SDI-12.....	78
Figura 47. Panel de inicio .....	80
Figura 48. Panel de visualización gráfica de variables .....	81
Figura 49. Panel de gestión base de datos .....	81

Figura 50. Archivo “.CSV” exportado de la interfaz .....	82
Figura 51. Ventana de confirmación del proceso que permite eliminar registros base de datos.....	83
Figura 52. Panel de visualización de imágenes .....	83
Figura 53. Módem celular Moxa OnCellG3151 .....	84
Figura 54. Configuración módem celular Moxa OnCellG3151.....	85
Figura 55. Indicadores de estado módem celular Moxa OnCellG3151.....	85
Figura 56. Presentación archivo “.MIS” .....	86
Figura 57. Estructura interna archivo “.MIS” .....	87
Figura 58. Estructura plantilla XML para Hydras3.....	88
Figura 59. Tipos de conexión unidad de control de potencia - PCU12.....	89
Figura 60. Conexión del calibrador como fuente de voltaje.....	92
Figura 61. Correlación de datos medidos.....	94
Figura 62. Generador de pulsos digitales.....	94
Figura 63. Verificación de pulsos leídos por el prototipo. ....	95
Figura 64. Conexión del analizador de protocolo SDI-12 utilizando Arduino UNO .....	96
Figura 65. Captura de pantalla de la trama de datos capturada por el analizador de protocolo SDI-12 .....	97
Figura 66. Verificación de lectura de datos SDI-12. ....	97
Figura 67. Diseño circuito placa base .....	98
Figura 68. Fabricación placa base .....	99
Figura 69. Módulo entrada SDI-12.....	99
Figura 70. Circuito final prototipo .....	100
Figura 71. Sensores hidrometeorológicos utilizados .....	100
Figura 72. Configuración inicial para la verificación de funcionamiento.....	101

Figura 73. Configuración de los sensores utilizados para la verificación de funcionamiento. .....	101
Figura 74. Adquisición de datos de temperatura con sensor ROTRONIC HygroClip HC2 .....	102
Figura 75. Adquisición de datos de humedad con sensor ROTRONIC HygroClip HC2..	102
Figura 76. Adquisición de datos de presión atmosférica con sensor VAISALA PTB100B .....	103
Figura 77. Adquisición de datos de precipitación con sensor RIMCO 8500 .....	103
Figura 78. Adquisición de datos velocidad del viento con sensor VAISALA WMT702....	104
Figura 79. Adquisición de datos dirección de viento con sensor VAISALA WMT702 .....	104
Figura 80. Archivos “.MIS” en el servidor FTP del IDEAM.....	105
Figura 81. Estructura del archivo “.MIS” recibido en el servidor del IDEAM. ....	106
Figura 82. Resultados adquisición y transmisión de temperatura del aire .....	107
Figura 83. Resultados adquisición y transmisión de humedad del aire .....	107
Figura 84. Resultados adquisición y transmisión de presión atmosférica.....	108
Figura 85. Resultados adquisición y transmisión de precipitación.....	108
Figura 86. Resultados adquisición y transmisión de velocidad del viento.....	109
Figura 87. Resultados adquisición y transmisión de dirección del viento .....	109
Figura 88. Gabinete Campbell ENC10/12 .....	110
Figura 89. Diseño estructura física.....	111
Figura 90. Elementos utilizados en la estructura física.....	112
Figura 91. Estructura física estación meteorológica .....	112
Figura 92. Interior del gabinete .....	113
Figura 93. Conexión Raspberry Pi - Módulo RTC1307 .....	129
Figura 94. Conexión Raspberry Pi – Conversor MCP3008 .....	136
Figura 95. Panel solar YingliSolar YL80P y características .....	145

## Índice de Códigos

Código 1. Captura de imágenes “Foto.py” .....	73
Código 2. Ejemplo del archivo de configuración XML .....	79
Código 3. Función crear archivo “.MIS” .....	87
Código 4. Prueba de entradas análogas con módulo MCP3008 en Python3 .....	138
Código 5. Plantilla XML utilizada para la integración de datos a Hydras3 .....	151
Código 6. Aplicación principal del prototipo desarrollado en Python3.....	152
Código 7. Librería de funciones creadas para el prototipo: “PrincipalDCP.py” .....	154
Código 8. Lectura de variables del archivo de configuración XML .....	158
Código 9. Programa en Arduino para la lectura de sensores SDI-12 .....	162

## INTRODUCCIÓN

A través de la historia, la humanidad se ha visto en la necesidad de construir herramientas con la capacidad de registrar y compartir todo tipo de información, con el fin de realizar investigaciones y desarrollar estudios, que permitan hacer análisis de los entornos y en base a los resultados, tomar decisiones operativas que ayuden a resolver problemas o necesidades en los diferentes campos en el que se desenvuelve el ser humano.

El estudio de las variables hidrometeorológicas ha sido de interés especial para toda la sociedad, ya que estas influyen directamente en todas las actividades humanas. El conocimiento de la climatología hace posible que se estudien a fondo problemáticas de tipo ambiental, buscando entender los diversos factores que aceleran fenómenos como el cambio climático y los desastres naturales, los cuales pueden llegar a afectar negativamente al medio ambiente, a la flora, a la fauna y en muchas ocasiones la vida de los seres humanos.

Las nuevas tecnologías en electrónica y telecomunicaciones, junto con diversas herramientas y aplicaciones a nivel de software, brindan posibilidades más eficaces y eficientes para el registro, procesamiento y transporte de la información en todo tipo de entornos en que se requieran acceso constante con el fin de realizar análisis que permitan obtener soluciones a diversas problemáticas. En Colombia, el campo de la meteorología e hidrología es de gran importancia debido a la riqueza natural que existe en todo el territorio nacional. Los estudios de las diferentes variables climatológicas en el país, en su mayoría, se realizan de forma convencional, utilizando instrumentos tradicionales en los que una persona observa las mediciones y registra la información de forma manual, lo cual implica una alta incertidumbre en la forma de medición, además de retardos en el proceso y análisis de la información.

Una de las necesidades tecnológicas en Colombia, es poder contar en todo su territorio con sistemas automáticos de monitoreo de variables hidrometeorológicas, que a diferencia de las observaciones realizadas por medio de estaciones convencionales, permitan recopilar datos confiables, verídicos y en tiempo real sin necesidad de desplazarse hasta el punto de medición, obteniendo información de calidad con la cual se

puede contribuir a la toma de decisiones acertadas en proyectos que favorezcan el desarrollo del país como lo son: la generación de energía eléctrica, la salud, la agricultura y la calidad del agua entre otros. Esto garantiza el manejo adecuado y aprovechamiento racional de los recursos naturales, permitiendo también entrar a la cultura de la prevención de desastres naturales, basado en generación de alertas tempranas, pronósticos y modelos de comportamiento climático.

Para el monitoreo general del país, el **Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM)**, actualmente cuenta con una red de estaciones hidrometeorológicas conformada por 2621 puntos de observación activos [1], de las cuales 2270 corresponden a estaciones de tipo convencional, 305 son estaciones automáticas con transmisión de datos y 46 son estaciones automáticas de almacenamiento *in situ*. En la Figura 1, se puede observar la distribución porcentual de la red de estaciones hidrometeorológicas del IDEAM.

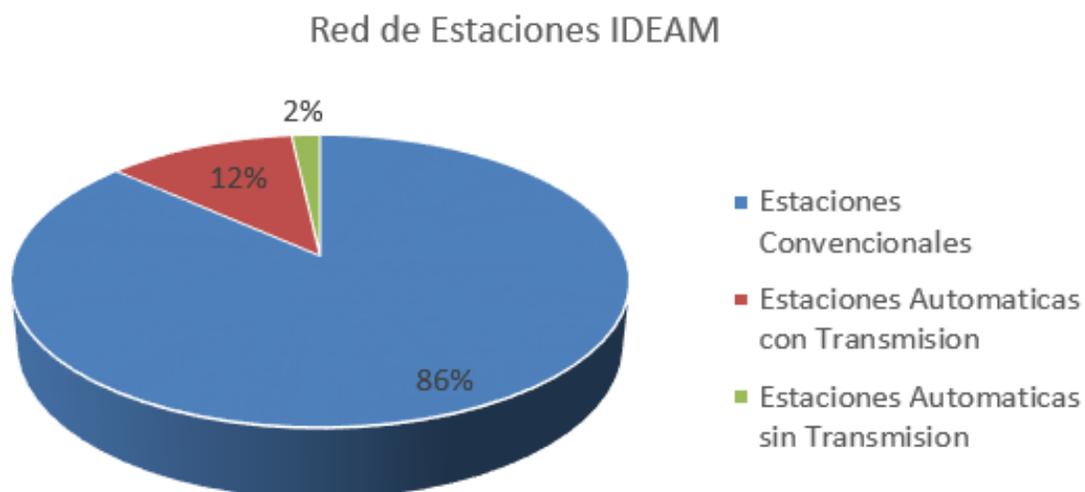


Figura 1. Clasificación de estaciones hidrometeorológicas en Colombia

Fuente: IDEAM

Actualmente en Colombia se están reemplazando los métodos de medición convencionales, por sistemas electrónicos automáticos, que permiten capturar información de sensores especializados y transmitirlos con una periodicidad horaria, sin embargo los dispositivos que actualmente realizan estas funciones son de fabricación

extranjera, lo cual implica que los costos sean extremadamente altos, por lo cual la cantidad de puntos de medición son escasos con respecto a la extensión del país.

El déficit de puntos de medición con transmisión en tiempo real, hace que la información no sea suficiente para realizar con precisión pronósticos climáticos y por ende prevenir por medio de alertas tempranas a poblaciones en riesgo de sufrir catástrofes naturales que eventualmente se puedan llegar a presentar, también se reduce la eficacia y veracidad de los estudios realizados en materia de energías renovables, salud, agricultura y calidad de agua entre otros.

# 1 OBJETIVOS

A continuación se presentan los objetivos planteados desde la etapa del anteproyecto:

## 1.1 Objetivo general

Diseñar e Implementar el prototipo de una plataforma colectora de datos (DCP), que permita adquirir información de sensores hidrometeorológicos especializados, almacenarla y transmitirla vía internet, integrándose al sistema nacional de visualización utilizado por el Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM).

## 1.2 Objetivos específicos

- Desarrollar por medio de un computador de placa reducida un sistema de bajo costo que permita la adquisición y procesamiento de datos de sensores hidrometeorológicos, el cual contará con hasta ocho (8) entradas análogas configurables en los rangos de 0-1V, 0-2.5V, y 0-5V, dos (2) entradas digitales por pulsos y una entrada serial con protocolo SDI-12.
- Almacenar la información de los valores adquiridos por los sensores junto con la fecha y hora de ocurrencia en una base de datos relacional.
- Implementar una interfaz gráfica de monitoreo local, que permita al usuario tener un entorno amigable de visualización y generación de gráficas de comportamiento de las diferentes variables.
- Integrar un sistema de cámara para tomar registros fotográficos del entorno cuando se excedan niveles críticos predeterminados en alguna de las variables medidas.
- Utilizar una comunicación inalámbrica GPRS que permita el envío de los datos vía internet desde el computador hacia el servidor del sistema nacional de visualización de estaciones automáticas utilizado en IDEAM.

## 2 METODOLOGÍA DE DESARROLLO

El proyecto se desarrolla con una metodología teórico-aplicativa, ya que basados en conceptos teóricos se busca la solución de una necesidad específica con el fin de implementar un sistema determinado como lo es el desarrollo de un sistema electrónico que permita la adquisición, transmisión y visualización de datos hidrometeorológicos.

Existen diversas técnicas para la realización de proyectos de ingeniería que involucran la implementación de hardware y software como se presenta en la referencia [2], específicamente, para el desarrollo del presente proyecto se decide implementar la metodología de desarrollo en cascada [3], la cual es un proceso secuencial dividido en 6 etapas relacionadas entre sí, y que describen el orden de ejecución con el cual se desarrolla el proyecto. Cada etapa tiene una serie de objetivos definidos y las actividades dentro de cada una, que contribuyen a la satisfacción de metas para la continuación a la etapa siguiente hasta la finalización del proyecto. Esta metodología tiene como gran ventaja la sencillez de ejecución y la consecución de objetivos de forma escalonada. En la Figura 2, se presentan las etapas del modelo de la metodología en cascada.

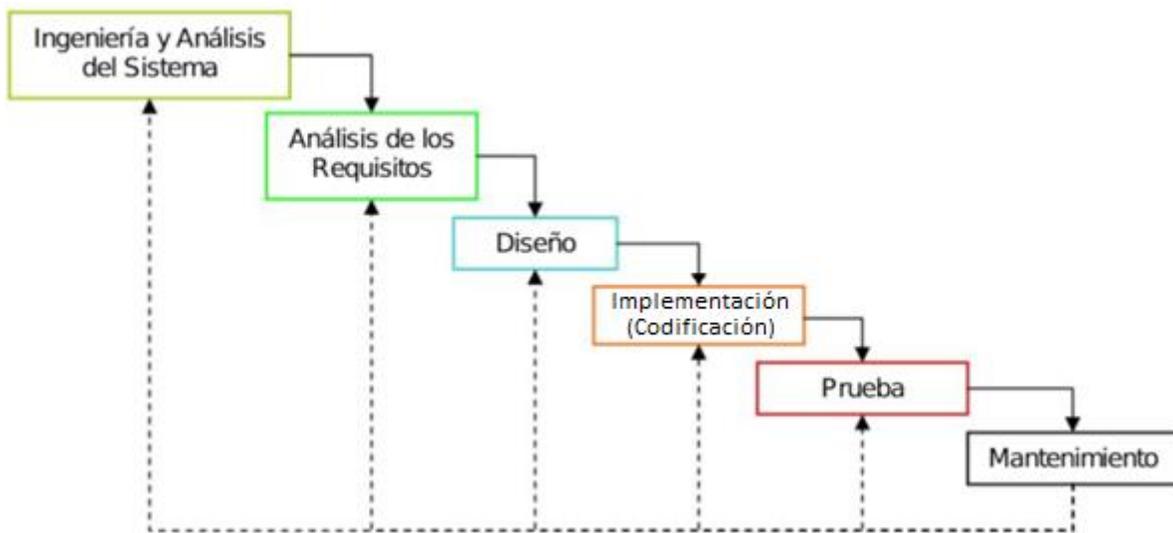


Figura 2. Etapas del Modelo en Cascada

Fuente:[3]

A continuación se describen cada una de las etapas de la metodología en cascada utilizadas durante la ejecución de este proyecto.

## **2.1 Ingeniería y Análisis del sistema**

En esta primera etapa se establecieron todas las necesidades del proyecto. Basados en la idea principal de construir el prototipo de una plataforma colectora de datos para sensores hidrometeorológicos especializados, se hace la revisión del estado del arte descrita en la **sección 3.5** y se hace una investigación general con el fin de verificar las tecnologías existentes aptas para el desarrollo del proyecto y los equipos que actualmente se encuentran en el mercado colombiano, adicionalmente se realizan consultas a profesionales del Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM) quienes brindan información del funcionamiento actual de las estaciones hidrometeorológicas convencionales y las necesidades tecnológicas para la implementación de estaciones automáticas.

## **2.2 Análisis de los Requisitos**

En esta etapa del proceso se evalúan las diferentes tecnologías y requisitos para la implementación del prototipo, y junto con el director del proyecto, se definen especificaciones técnicas básicas como la cantidad de entradas para los diferentes tipos de sensores (análogos, digitales y seriales con protocolo SDI-12), el almacenamiento de información en una base de datos relacional, las funcionalidades de la interfaz gráfica de monitoreo, el método y periodicidad para la captura de imágenes fotográficas, así como las características del sistema de transmisión celular GPRS, y los requerimientos necesarios para realizar la integración de información al sistema nacional de visualización del IDEAM, los cuales son la base para la construcción de los objetivos general y específicos presentados en la sección **1 (OBJETIVOS)** y con los que además se plantea el diseño utilizado para la implementación del prototipo.

## 2.3 Diseño

Una vez analizados las diferentes tecnologías para el desarrollo del prototipo, y teniendo en cuenta los requerimientos y características de funcionamiento y operación definidos en la etapa de análisis, se propuso un diseño descriptivo, basado en componentes electrónicos de hardware y herramientas software, lo cual permite dar cumplimiento a los objetivos planteados. En la selección de cada uno de los componentes a utilizar se tuvo en cuenta, la funcionalidad, costo y compatibilidad con los demás elementos, a fin de garantizar el correcto funcionamiento de todos los dispositivos y la posibilidad de integrar hardware de manera ágil. Así mismo uno de los componentes más importantes del diseño es el uso de herramientas de software libre para así eliminar costos asociados a licencias o permisos de uso y de esta forma poder implementar y modificar de manera independiente el código fuente de la solución planteada. En la Tabla 1, se presentan las etapas que integran el diseño del prototipo junto a los componentes de hardware o software que hacen parte de estas.

<b>ETAPAS</b>	<b>COMPONENTES</b>
Configuración del reloj en tiempo real	RTC - DS1307
Adquisición y procesamiento de datos	Rapsberry Pi Model 2
Medición de sensores digitales	Pines GPIO Rapsberry Pi
Medición de sensores análogos	ADC – MCP3008
Medición de sensores seriales sdi-12	Arduino UNO y Librería SDI-12
Almacenamiento en base datos	MySQL
Interfaz gráfica de visualización local	Python y C#
Adquisición de imágenes	Cámara RP 5Mpx
Sistema de comunicación inalámbrica	Modem Celular MOXA OneCell G3151
Interfaz de monitoreo remoto	Servidor FTP e Hydras3
Sistema de alimentación	Regulador PCU-12

Tabla 1. Etapas y componentes del diseño del prototipo.

Cada una de las etapas de diseño se conecta entre sí obteniendo un producto definitivo, el cual es el prototipo de una plataforma colectora de datos (DCP), que permita adquirir información de sensores hidrometeorológicos especializados, almacenarla y transmitirla vía internet, integrándose al sistema nacional de visualización del IDEAM. En la Figura 3, se puede observar un diagrama de bloques en el cual se muestran elementos de diseño que componen el prototipo implementado.

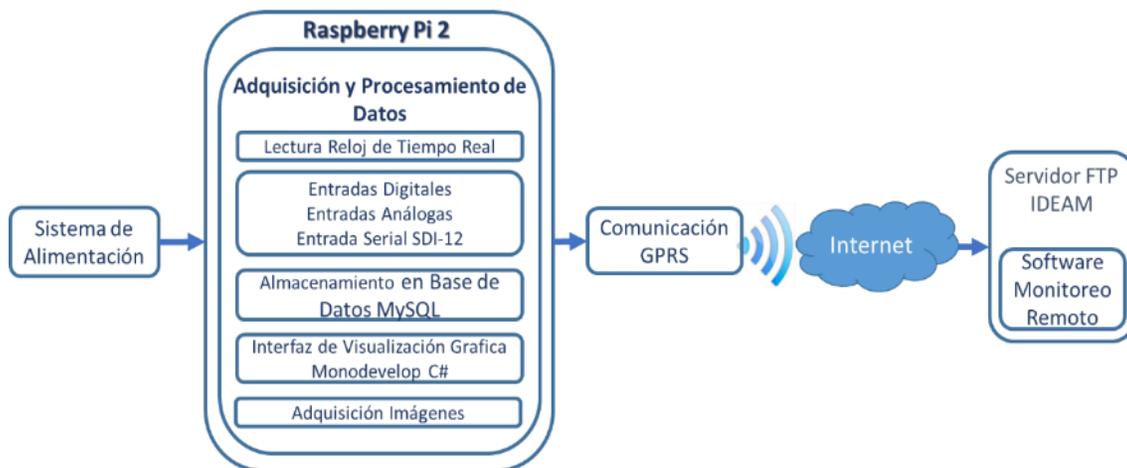


Figura 3. Diagrama de bloques diseño del prototipo

## 2.4 Implementación (codificación)

Una vez definido el diseño, se procedió con la implementación del prototipo, integrando los diferentes componentes de hardware y los acondicionamientos necesarios para el funcionamiento en específico de la tarjeta Rapsberry Pi, el reloj en tiempo real, el conversor análogo digital y la pantalla LCD, para lo cual se realizaron los diferentes montajes de los circuitos correspondientes un placas de prueba para después proceder con la codificación o implementación del software.

La codificación para el desarrollo del software implementado en el prototipo, se utilizaron diversas herramientas de uso libre como el motor de base de datos MySQL, en el que se realiza el almacenamiento de información, Python3, con la cual se gestiona la adquisición y procesamiento de datos dentro de la tarjeta Raspberry Pi, y finalmente se utiliza el entorno de desarrollo integrado MonoDevelop y su compilador de C#, para el desarrollo de la interfaz gráfica de usuario para la visualización de los valores adquiridos por los sensores, la configuración de la captura de imágenes, y la gestión sobre la base de datos. En el capítulo 4 (**DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO**) se explica en detalle el desarrollo y codificación del diseño planteado.

## **2.5 Prueba**

Luego de realizar la implementación física de los componentes electrónicos y la instalación y configuración de las diferentes herramientas de software cada una de las etapas del prototipo, se somete a pruebas de funcionamiento y operación, por ejemplo verificando el adecuado registro de información en el sistema de almacenamiento de datos, la operación y manejo de la interfaz gráfica, el formato y captura en el sistema de adquisición de imágenes, y la coherencia y visualización en el sistema de transmisión de datos al software de monitoreo remoto. En la sección **5 (VERIFICACIÓN DE FUNCIONAMIENTO)** se describe en detalle las pruebas realizadas para comprobar el adecuado comportamiento del sistema de adquisición de datos por medio de las entradas análogas, digitales y seriales SDI-12.

## **2.6 Mantenimiento**

Una vez realizadas las diferentes pruebas de funcionamiento se contempla la etapa de mantenimiento, la cual hace referencia a los cambios o adaptaciones que se realicen en cada una de las etapas metodológicas del prototipo debido a errores encontrados, problemas de compatibilidad o mejoras que surgen a medida que el desarrollo del prototipo se va ejecutando.

### **3 MARCO REFERENCIAL**

En esta sección se hace referencia a las observaciones meteorológicas, la Organización Meteorológica Mundial y los dos tipos de observaciones superficiales existentes a nivel mundial; las estaciones convencionales y las estaciones automáticas. Posteriormente se dará un enfoque local de como actualmente funcionan estas observaciones en Colombia, los beneficios del uso de las observaciones meteorológicas en diferentes campos, seguido de la revisión del estado del arte y la comparación de tecnologías existentes en el mercado actual.

#### **3.1 Observaciones meteorológicas a nivel mundial**

Teniendo en cuenta la información obtenida de [4], se define una observación meteorológica como la medición o determinación, de las condiciones atmosféricas, utilizando la vista o algún instrumento adecuado en un momento dado y en un determinado lugar.

A través del tiempo los seres humanos alrededor del mundo se han dado cuenta de la importancia de las observaciones meteorológicas para la productividad de sus diferentes actividades, por esta razón diferentes naciones se vieron en la necesidad de estandarizar las mediciones realizadas a las diferentes variables hidrometeorológicas. De esta necesidad nace la Organización Meteorológica Internacional (OMI), la cual se fundó en 1873 y con el pasar de los años, en 1950 se convirtió en la Organización Meteorológica Mundial (OMM), la cual hoy en día es el organismo especializado reconocido por las Naciones Unidas para la meteorología, hidrología, y ciencias geofísicas [5]. Esta organización tiene dentro de sus objetivos la cooperación para la creación de redes de observaciones meteorológicas, fomentando la colaboración entre los diferentes Servicios Meteorológicos e Hidrológicos Nacionales (SMHN), facilitando así, el intercambio gratuito y sin restricciones de datos, información, productos y servicios, en favor del bienestar y protección de la sociedad mundial.

La OMM a través de sus diferentes programas contribuye en la formulación de políticas, normas, guías, procedimientos y recomendaciones de impacto nacional e internacional, en todos los temas relacionados con meteorología, hidrología y ciencias geofísicas, y además desempeña una destacada función en las actividades destinadas a vigilar y proteger el medio ambiente, contribuyendo con el desarrollo y bienestar de todos los países miembros.

La OMM ha establecido el programa del Sistema Global de Observaciones (GOS), el cual comprende estaciones de observación en tierra, en el mar, en el aire y en el espacio exterior, convirtiéndose en el programa más importante de la OMM para la observación, registro y presentación de informes sobre el tiempo, el clima, el medio ambiente, la predicción operativa y servicios de alerta e información relacionada. Los componentes del GOS son propiedad de cada uno de los países y la usabilidad y operación está a cargo de cada uno de los SMHN. [6].

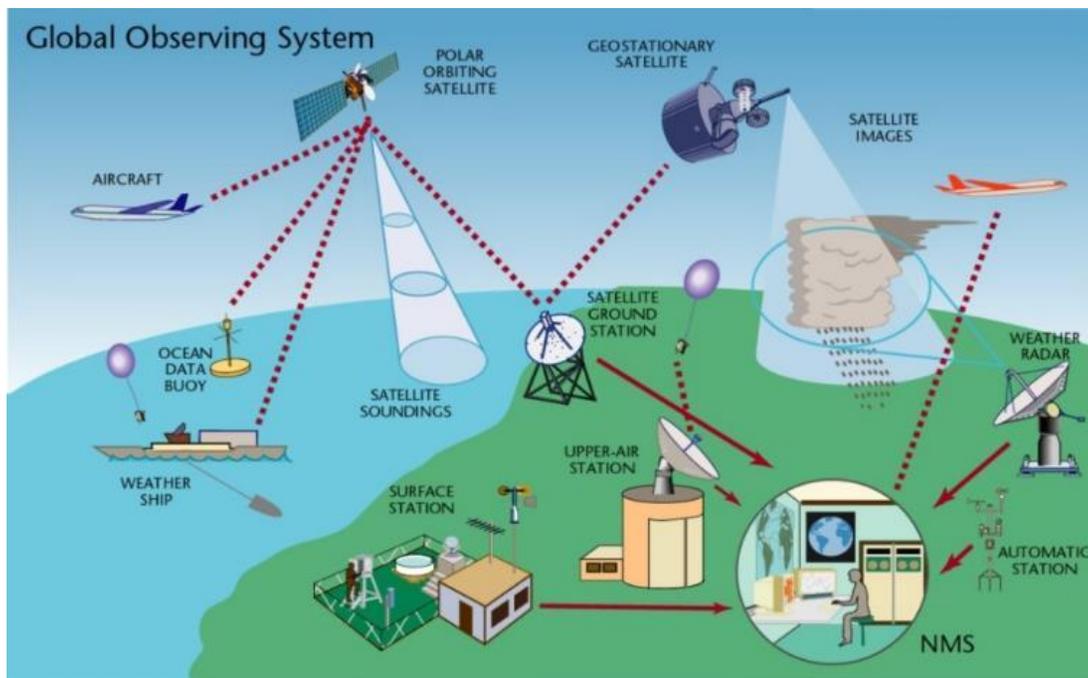


Figura 4. Sistema Global de Observación (GOS)

Fuente: [www.wmo.int](http://www.wmo.int)

Dentro de los componentes más importantes del Sistema Global de Observaciones están las observaciones con radares meteorológicos, observaciones de satélites

geoestacionarios, observaciones de altura, observaciones marítimas, observaciones provenientes de aviones, y aquellas que serán objeto de estudio en este documento, las observaciones de superficie. En la Figura 4 se pueden ver los componentes generales del Sistema Global de Observaciones.

En el marco del decimoquinto congreso meteorológico mundial del año 2007, se estableció la iniciativa para crear el Sistema mundial integrado de sistemas de observación de la OMM (WIGOS) [7], el cual tiene como objetivo: *“Reforzar la capacidad de los Miembros para adquirir, desarrollar, aplicar y utilizar sistemas de superficie y espaciales integrados y compatibles para efectuar observaciones meteorológicas climáticas e hidrológicas así como otras observaciones medioambientales conexas sobre la base de normas mundiales fijadas por la OMM y organizaciones asociadas”* [8].

Generalmente los sistemas de observación de la OMM han sido implementados por separado, lo cual ha traído consigo una serie de incompatibilidades, deficiencias, esfuerzos duplicados, y en general mayores costos. En la actualidad existe una creciente demanda de datos, productos y servicios hidrometeorológicos, que los sistemas de observación tradicionales no pueden satisfacer. Con el sistema mundial integrado de sistemas de observación, se podrán solventar los inconvenientes mejorando en tecnología, capacidad, eficiencia, calidad y fiabilidad de las observaciones realizadas a nivel mundial, permitiendo así una estandarización de la información y colaboración mutua para los miembros de la OMM y las diferentes entidades externas interesadas [9].

### **3.2 Observaciones de superficie**

De acuerdo con el manual del sistema mundial de información [10], las observaciones de superficie se hacen en lugares o espacios dedicados a realizar mediciones de variables atmosféricas, generalmente conocidas como estaciones meteorológicas, las cuales deben estar dotadas de instrumentos fiables y precisos, debidamente calibrados a fin de satisfacer las necesidades del tipo de medidas hidrometeorológicas deseadas.

En este manual [10] se especifican los diversos requisitos a nivel mundial, regional y nacional que debe satisfacer una red de estaciones. En este documento se explican los

diferentes tipos de estaciones y elementos que las componen, según las aplicaciones particulares. De acuerdo con la guía de métodos y observación [11], en una estación que realiza mediciones de superficie, generalmente (dependiendo del tipo) se observan los siguientes parámetros:

- Tiempo presente
- Tiempo pasado
- Dirección y velocidad del viento
- Nubosidad
- Tipo de nubes
- Altura de la base de las nubes
- Visibilidad
- Temperatura
- Humedad relativa
- Presión atmosférica
- Precipitación
- Capa de nieve
- Insolación y/o radiación solar
- Temperatura del suelo
- Evaporación
- Nivel (ríos y mares)

Las observaciones de superficie se realizan básicamente por medio de dos tipos de estaciones, observaciones marítimas y observaciones terrestres, las cuales a su vez se pueden clasificar en estaciones de superficie dotadas de personal (convencionales) y estaciones de superficie automáticas. A continuación se realiza la descripción de estos tipos de estaciones terrestres.

### **3.2.1 Observación en estaciones dotadas de personal (Convencionales)**

En este tipo de observaciones las mediciones de diferentes variables climatológicas se realizan de forma tradicional, utilizando instrumentos convencionales, por ejemplo, en guías técnicas especializadas [11] [12], se mencionan elementos como termómetros líquidos en capsulas de vidrio para la temperatura, barómetros de mercurio para la presión atmosférica, anemómetros de cazoletas y veletas para la velocidad y dirección del viento, pluviómetros ordinarios para las precipitaciones, o regletas de nivel (limnímetros) para ríos, lagos y embalses, entre otros. En la Figura 5, se puede observar algunos de los instrumentos convencionales para la medición de variables meteorológicas.

Para su funcionamiento, las estaciones convencionales dependen de la intervención humana, donde una persona (observador de superficie), es el responsable de registrar las

mediciones y además se encarga de vigilar las condiciones de la superficie (por ejemplo crecimiento del césped), se ocupa del mantenimiento básico de los instrumentos, y de examinar daños o detectar quebrantos a la seguridad [13].

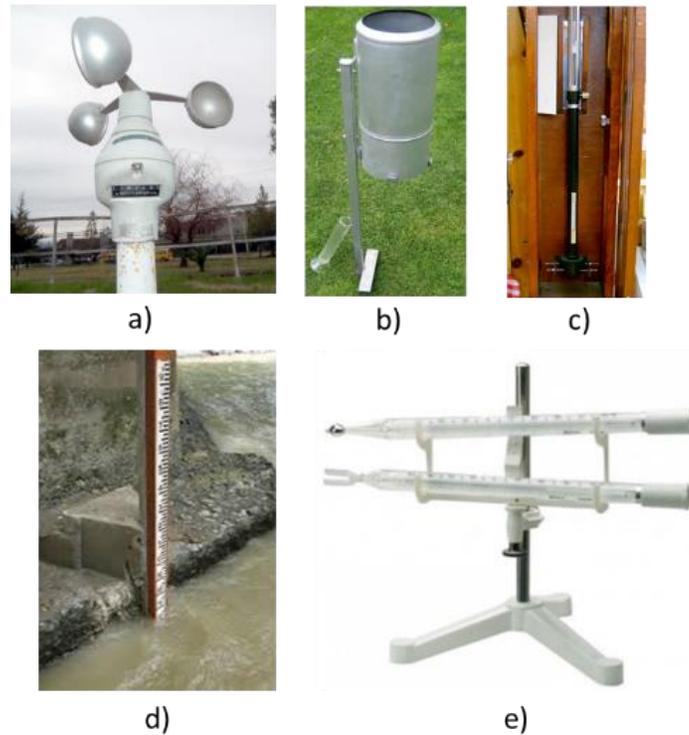


Figura 5. Instrumentos convencionales

a) Anemómetro, b) Pluviómetro, c) Barómetro, d) Limnómetro, e) Termómetro

El gran inconveniente de las observaciones con estaciones convencionales es que de acuerdo a la recomendación de la OMM [10], el observador de superficie debería realizar la medición, cada 3 o 6 horas, en itinerarios correspondientes a las 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00 y 21:00 en tiempo coordinado universal (UTC). Adicionalmente y tal como se indica en [11], el observador debe registrar el valor medido en un cuaderno de notas de forma manual, a medida que se van obteniendo.

Las mediciones con instrumentos convencionales hacen que se presenten errores humanos en la toma de datos dependiendo de factores como la regularidad de la medición, el ángulo de visión o la sobriedad del observador de superficie. Con el fin de mitigar este tipo de errores, en [11] se hace referencia a instrumentos de registro

autográfico, como lo son: termógrafos, anemógrafos, heliógrafos, pluviógrafos o limnógrafos, los cuales por medio de una hoja de papel previamente grabada con la escala adecuada, dibujan el comportamiento de la variable a través de una plumilla con tinta accionada mecánicamente, sin embargo este tipo instrumentos requiere el cambio periódico de la papelería por parte de un observador. En la Figura 6, se observa algunos instrumentos autográficos.

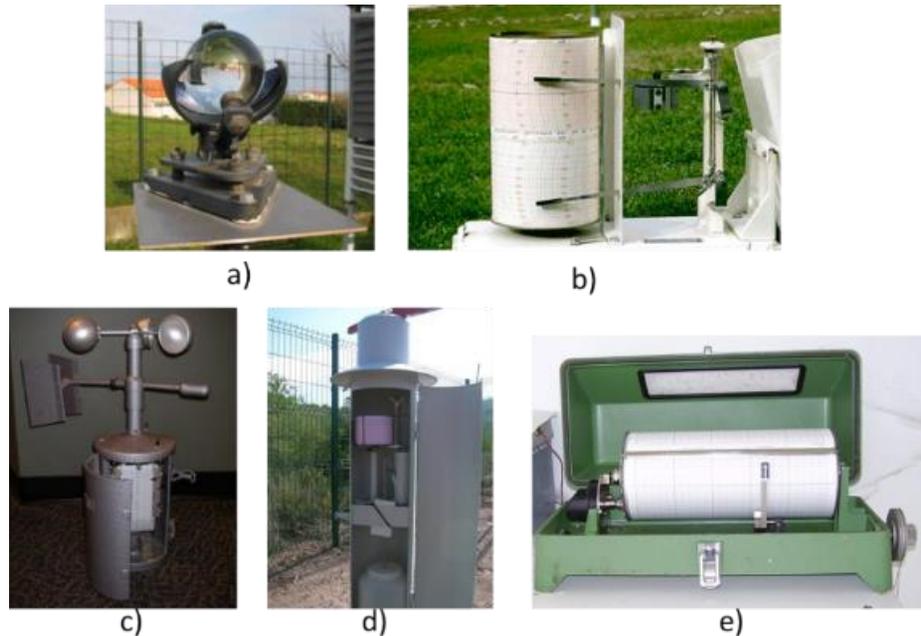


Figura 6. Instrumentos de registro autográfico

a) Heliógrafo, b) Termohigrógrafo, c) Anemógrafo, d) Pluviógrafo, e) Limnógrafo

Toda la información tanto de gráficas como de libretas, queda en poder del observador de superficie, a la espera de ser recogida por el personal encargado, llegando a ser analizada por los expertos meteorólogos en la mayoría de las ocasiones meses después, lo cual hace que esta forma de observación sea ineficiente para la generación de alertas tempranas ante fenómenos o catástrofes repentinas.

### 3.2.2 Observación de datos de forma automática

De acuerdo con [14], una estación meteorológica automática es aquella “*en la que se realizan y transmiten observaciones automáticamente*”, este tipo de estaciones sirven

para aumentar el número y la fiabilidad de las observaciones de superficie, ya que estas permiten reducir los errores humanos, proporcionando datos de manera constante y continua, aumentando la densidad de la red existente a lugares de difícil acceso, mediante el uso de tecnologías modernas y técnicas de medición digitales. Este tipo de estaciones está compuesta generalmente de tres partes: percepción de variables, procesamiento de información, y transmisión de datos (telemetría).

La percepción de variables es la etapa en la que se utilizan sensores electrónicos como los mostrados en la Figura 7, los cuales basados en principios físico-químicos transforman las variables en señales análogas de voltaje, corriente, resistencia o en señales digitales seriales o de pulsos.

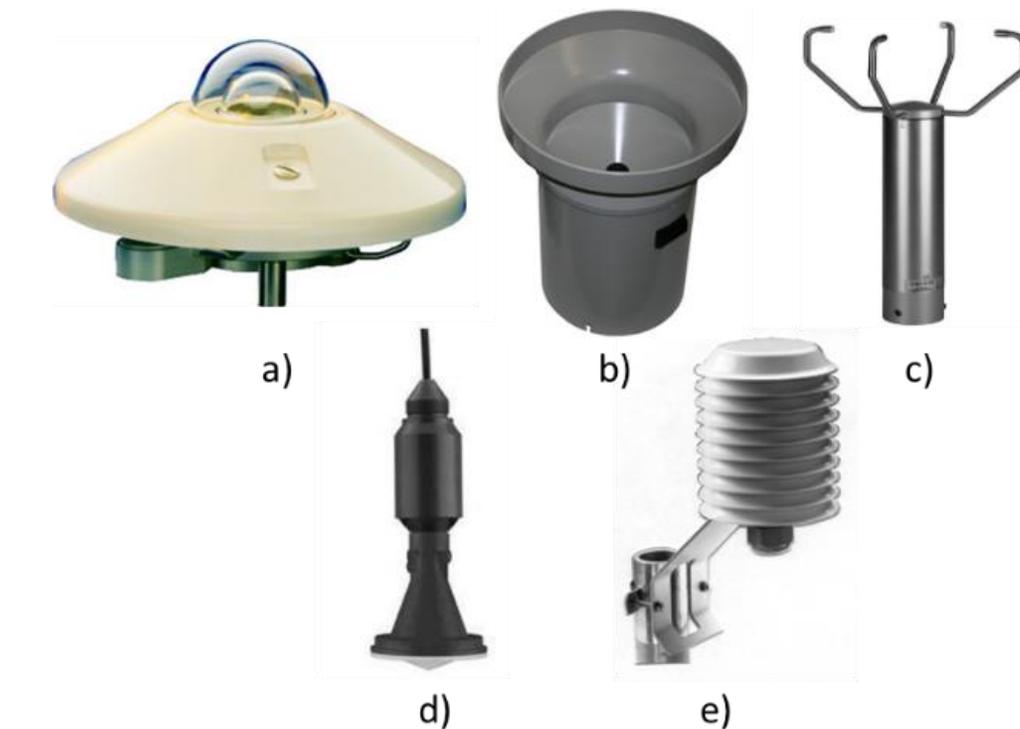


Figura 7. Sensores electrónicos

a) Radiación Solar, b) Precipitación, c) Dirección y Velocidad del Viento, d) Nivel, e) Temperatura y Humedad del Aire

Para el procesamiento de información se utiliza un dispositivo que de acuerdo con [11], es el elemento más importante de una estación meteorológica automática, el cual es

conocido como unidad central de procesamiento, plataforma colectora de datos, DCP por sus siglas en inglés (*Data Colector Plataforma*) o *datalogger*. Este es el encargado de realizar la adquisición de las señales eléctricas provenientes de los sensores, por medio de canales entrada/salida que permiten convertir estas señales de manera proporcional en un formato legible por un microprocesador, para que sea almacenada en una memoria y prepararla para la transmisión de datos. En la Figura 8 se pueden observar imágenes de algunas plataformas colectoras de datos disponibles en el mercado. En la sección 3.5, se realiza la comparación de características técnicas de algunas de estas.

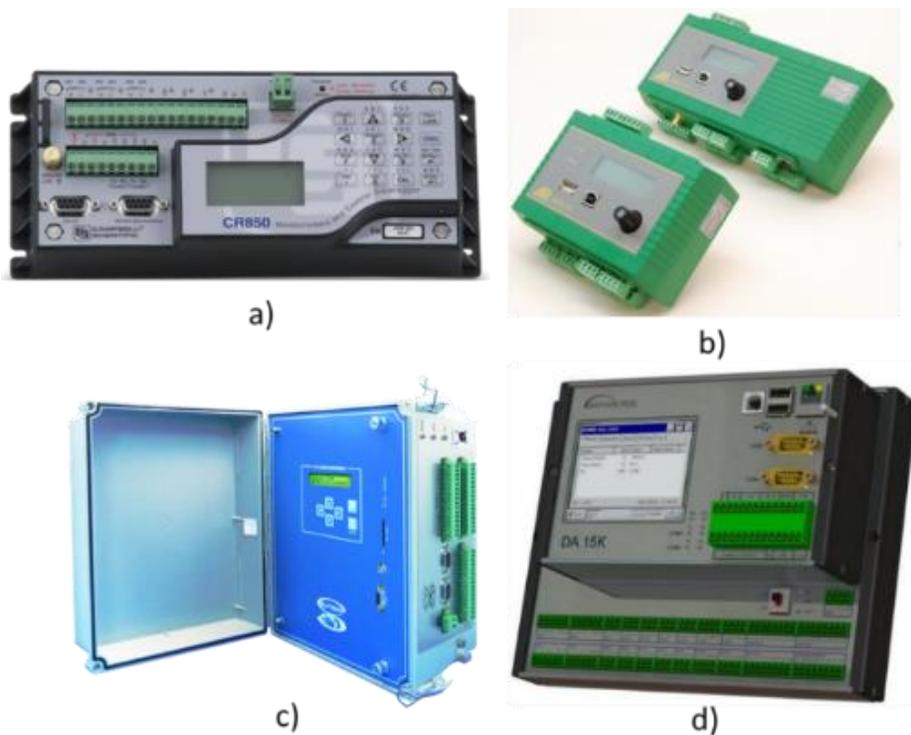


Figura 8. Plataformas colectoras de datos disponibles en el mercado  
a) Campbell, b) OTT, c) Sutron, d) Siap+Micros

Las estaciones automáticas no siempre cuentan con sistemas de transmisión de datos, ya que es posible encontrar estaciones que únicamente almacenen información “*in situ*”. Sin embargo, dependiendo de la aplicación es importante contar con dispositivos que permitan realizar un monitoreo remoto, con lo cual el operador podrá recibir información de los sensores almacenados en el DCP. En [15] se describen diferentes tipos de telemetría, como por ejemplo: Radio, Módem, Telefonía celular, y Satélite. En la Figura 9,

se pueden observar diferentes tipos de telemetría para estaciones meteorológicas automáticas.

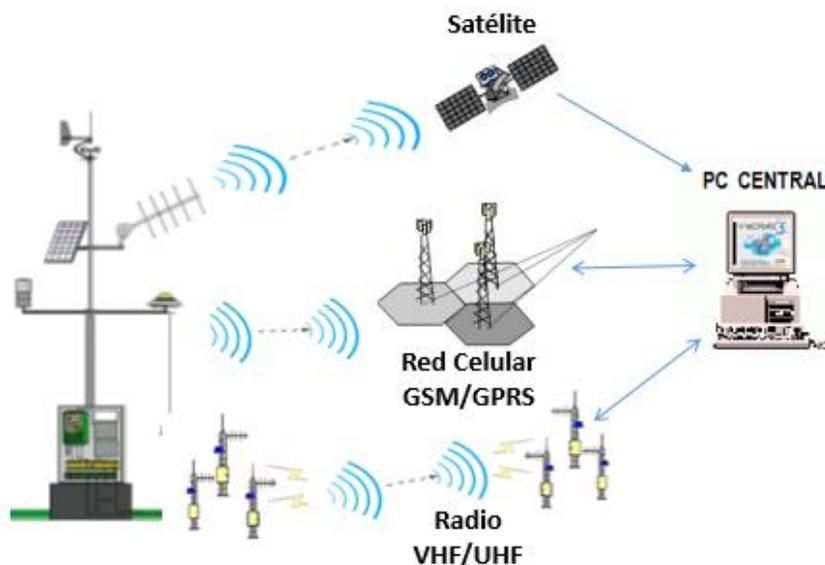


Figura 9. Tipos de telemetría en estaciones meteorológicas automáticas

Fuente: IDEAM

### 3.3 Observaciones meteorológicas en Colombia

Colombia es un país miembro de la Organización de las Naciones Unidas (ONU), por lo cual se acoge al sistema global de observaciones establecido por la OMM, siendo su principal forma de recopilación de datos climáticos las observaciones terrestres a través de estaciones de superficie. A continuación se presentan algunas de las entidades que realizan observaciones hidrometeorológicas en Colombia.

#### 3.3.1 Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM)

En 1969 se crea el Servicio Colombiano de Meteorología e Hidrología (SCMH), dando inicio a la red básica nacional de estaciones hidrometeorológicas, con lo cual se centraliza la información que de manera dispersa se generaba en el país. En 1976, con la reestructuración del SCMH, las funciones relacionadas con la red meteorológica pasan a cargo del Instituto de Hidrología, Meteorología y Adecuación de Tierras (HIMAT), y

posteriormente en 1993, por medio de la ley 99 de ese año se crea el Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM), quien actualmente tiene como misión básica la generación de información sobre el estado de los recursos naturales y las condiciones hidrometeorológicas del país, para la toma de decisiones por parte de actores y usuarios, los cuales pueden ser: autoridades, planificadores, academia, comunidades y sectores económicos y sociales [16].

En la Tabla 2 se observa la clasificación de las estaciones hidrometeorológicas que actualmente se encuentran activas dentro del Catálogo Nacional de Estaciones (CNE) del IDEAM [1], existen diferentes tipos de estaciones de superficie que el IDEAM categoriza dependiendo de sus variables y formas de medición creando una red de estaciones de observación y registro de diferentes fenómenos atmosféricos. Basado en la recopilación de datos hidrometeorológicos que se obtiene de la red de estaciones que se encuentran por todo el territorio nacional, el IDEAM genera información climática de interés para todo el país.

<b>Categoría</b>	<b>Sigla</b>	<b>Cantidad</b>
Pluviómetrica	PM	1166
Linimétrica	LM	308
Climatológica Ordinaria	CO	269
Linigráfica	LG	350
Hidrológica Automática	HA	38
Sinóptica Principal	SP	42
Climatológica Principal	CP	209
Pluviográfica	PG	123
Mareográfica R	MR	1
Agrometeorológica	AM	61
Meteorológica Especial	ME	36
Sinóptica suplementaria	SS	7
Radio Sonda	RS	4
Mareográfica	MM	5
Presión de Poros	DEH	2
<b>TOTAL:</b>		<b>2621</b>

Tabla 2. Clasificación de estaciones hidrometeorológicas en Colombia

Fuente: IDEAM

### 3.3.2 Otras entidades a nivel nacional

Además del IDEAM existen diferentes entidades en el país tanto de carácter público como privado que utilizan estaciones hidrometeorológicas para realizar observaciones a diferentes tipos de variables de interés específico.

En Colombia existen 33 **Corporaciones Autónomas Regionales y de Desarrollo Sostenible (CAR)** las cuales son entidades corporativas de carácter público, integradas por los entes territoriales [17]. Estas corporaciones están encargadas de administrar dentro de su jurisdicción, el medio ambiente, los recursos naturales renovables y no renovables, además de propender por el desarrollo sostenible de su área. Algunas de las CAR, para un mejor desarrollo de sus actividades, manejan programas propios de meteorología y calidad de agua a través de sistemas de monitoreo y gestión de datos.

**El Centro de Innovación Tecnológica del Sector Floricultor (CENIFLORES)** utiliza información de datos meteorológicos de 18 estaciones de la sabana de Bogotá y 4 de Antioquia, los cuales se utilizan para conocer la distribución territorial de la precipitación, radiación solar, temperatura, velocidad del viento y humedad del aire en esas zonas.

De modo similar al anterior, la **Federación Nacional de Arroceros, (FEDEARROZ)**, tiene instalando una red de 25 estaciones meteorológicas automáticas en las principales zonas productoras del país.

La **Corporación de Investigaciones Agropecuarias de Colombia, (CORPOICA)**, ha instalado estaciones agrometeorológicas automáticas en sus diferentes Centros de Investigación Agropecuaria, localizados en las principales zonas agrícolas del país, con el fin de recolectar datos meteorológicos necesarios en los proyectos de investigación que desarrolla.

### 3.4 Beneficios de las observaciones meteorológicas

El tener un sistema de observación de variables hidrometeorológicas, favorece de gran manera a toda la sociedad en múltiples campos, como pueden ser todos los relacionados

con los desastres naturales, suministros de alimentos, el comportamiento del agua y a la generación de energía, los cuales son factores cada vez más importantes en la conformación de la economía mundial y del bienestar de un país. A continuación se describen algunos de los campos beneficiados con las observaciones Hidrometeorológicas.

### **3.4.1 Prevención de desastres naturales**

Hoy en día es innegable que el medio ambiente en general se ha ido deteriorando, producto del cambio climático y la destrucción de la biodiversidad por parte del hombre. A nivel mundial se presentan con mayor frecuencia fenómenos como terremotos, erupciones volcánicas, huracanes, deslizamientos, incendios e inundaciones, los cuales generan catástrofes que destruyen infraestructuras, alteran la economía, y toman gran cantidad de vidas año tras año.

De acuerdo con [18] dentro de los desastres naturales más comunes, están los de tipo meteorológico, climatológico, geofísico e hidrológico, los cuales a nivel mundial, para el periodo 2004-2013 reportaron un promedio anual de 348 eventos, dejando en igual periodo de tiempo un promedio de 199.2 millones de víctimas entre mortales y afectados. Las estadísticas desde 1980 hasta 2014, por eventos de pérdida a causa de desastres naturales [19], muestran un comportamiento incremental con el pasar de los años, siendo eventos de tipo climatológico e hidrológico los que más número de acontecimientos presentan, lo cual concuerda con [18], donde se indica que para la última década (1995-2015), el mayor porcentaje de eventos relacionados con desastres naturales lo tienen las inundaciones con un 43%, seguido por otros sucesos como, tormentas 28%, terremotos 8%, temperaturas extrema 6%, deslizamientos 5%, sequias 5%, incendios 4%, y actividades volcánicas con un 2%.

Para el año 2014, alrededor de 99 países reportaron 324 eventos sobre desastres naturales, destacándose, China con 40, Estados Unidos con 20, India con 15, Japón con 13 y Filipinas con 13, durante ese año se presentaron 7823 personas muertas por estas calamidades, siendo China (1441), India (1037), Nepal (588), Afganistán (575) y Perú (554), los que más víctimas mortales presentaron; alrededor del mundo las pérdidas

materiales ascienden a U\$ 99.2 billones de dólares. Los cinco países que más generaron este tipo de pérdidas son: China con U\$28.95 Billones, India con U\$23.36 Billones, Estados Unidos con U\$16.78 Billones, Japón con U\$6.58 Billones, y Brasil con U\$5.20 Billones [20].

A pesar de que Colombia no aparece dentro de los primeros lugares de países con eventos, mortalidad o pérdidas materiales a causa de desastres naturales, cuenta con un número importante de calamidades. Según la Unidad Nacional para la Gestión del Riesgo de Desastres [21], en Colombia durante el año 2014 se atendieron más de 2900 emergencias afectando directamente a centros poblados, dejando más de 1.200.000 personas afectadas, de las cuales más del 60% fueron producto de catástrofes naturales como inundaciones, deslizamientos, vendavales e incendios forestales, entre otros. Todos estos han dejado víctimas mortales, personas desaparecidas y muchas familias perjudicadas. En la Tabla 3, se presenta un resumen de las emergencias medioambientales presentadas en Colombia durante el 2014.

	AFECTACION				
	MUERTOS	HERIDOS	DESAPA.	PERSONAS	FAMILIAS
AVALANCHA	1	14	2	5859	1167
CONTAMINACION	2	5	0	813	161
CRECIENTE SUBITA	5	4	2	1773	382
DESLIZAMIENTO	69	188	7	563491	74299
EROSION	0	0	0	1515	303
GRANIZADA	5	10	0	2009	436
INCENDIO FORESTAL	2	5	0	340	48
INUNDACION	86	238	19	119686	22031
MAR DE LEVA	0	0	0	0	0
OTROS	1	14	0	634	135
SEQUIA	6	27	0	17087	3407
SISMO	2	30	0	45	3
TORMENTA ELECTRICA	4	0	0	859	232
VENDAVAL	67	162	14	72668	15216

Tabla 3. Consolidado de atención a emergencias por eventos medioambientales 2014

En Colombia, el antecedente más importante de catástrofe natural, es la tragedia ocurrida el 13 de noviembre de 1985, producto de la erupción del volcán nevado del Ruiz, que ocasiono un deshielo en el glaciar, generando una avalancha de lodo y piedra, que se

encamino por los cauces de los ríos Lagunillas, Combeima, Chinchiná y Gualí afectando a los departamentos de Caldas y Tolima, principalmente a la población de Armero, dejando víctimas mortales de más de 23000 personas, [22].

Recientemente entre la noche del 31 de Marzo y la madrugada del 1 de Abril de 2017, la capital del departamento de Putumayo, Mocoa, sufre una terrible avalancha producto del desbordamiento de los tres ríos que atraviesan el municipio, el Mocoa, el Mulato y el Sangoyaco, arrastrando gran cantidad de piedras y lodo por 17 barrios, dejando más de 300 víctimas mortales [23]. Según el reporte del IDEAM ese día se presentó un volumen de precipitación de 129,3mm, lo cual es el equivalente al 30% del total de lluvias que normalmente caen durante todo el mes en esa zona [24].

El tener un sistema de observación de variables hidrometeorológicas, beneficia de gran manera a la comunidad ya que con la ayuda de este se puede generar un acceso rápido a pronósticos del tiempo, alertas tempranas, datos sobre las condiciones de la tierra y el mar, y se puede reducir la pérdida de vidas y elementos materiales en casos de desastres naturales y así fortalecer la preparación para enfrentar los desastres, para su predicción y respuesta, [25].

### **3.4.2 Generación de energía de fuentes renovables**

Uno de los campos en los que más se favorece la sociedad con las observaciones hidrometeorológicas, es en el desarrollo de estudios sobre las energías renovables. Basados en [26], a finales del año 2014 la capacidad global de generación de energía hidroeléctrica alcanzo los 1055GW, en donde solo China aportó el 27% del total y junto con Brasil, Estados Unidos, Canadá, Rusia e India representan el 60% de la capacidad total. De acuerdo con [27], hacia el año 2000, la energía hidroeléctrica constituía el 93% de las energías renovables en el mundo, sin embargo en 2014 ocupa el 64%, debido al rápido crecimiento que desde 2010 ha tenido la energía solar fotovoltaica que paso de una capacidad de generación global de 40GW en 2010 a 177GW en 2014.

Otro tipo importante de energía renovable es la energía eólica, la cual también ha tenido un crecimiento exponencial en los últimos años y que de acuerdo con [28], genera el 3%

del total de energía eléctrica en el mundo, países como China, Estados Unidos, Alemania, España e India lideran el ranking de capacidad de generación de este tipo de energía, sin embargo para países como Dinamarca, Nicaragua, Portugal y España, la energía eólica representa más del 20% de la electricidad generada en cada país.

En Colombia la generación de energía eléctrica [29], es producida en un 70% por plantas hidroeléctricas, mientras que alrededor del 29% es generada por plantas térmicas fósiles, dejando apenas cerca del 1% para la generación por medio de otro tipo de energías, a pesar de su rápido crecimiento a nivel mundial energías como la solar y la eólica, son poco utilizadas y todavía tienen muchos espacios por explorar convirtiéndolas en una oportunidad interesante para el desarrollo económico y social del país.

Las observaciones hidrometeorológicas son de vital importancia para realizar el monitoreo y pronóstico de las capacidades de generación de electricidad en la energía hidroeléctrica, solar, marina, y eólica entre otras, lo cual permite efectuar evaluación a los impactos ambientales, el consumo, los riesgos climáticos así como la adecuación de la oferta a la demanda de energía.

### **3.4.3 Prevención de enfermedades**

La salud es otro campo beneficiado de las observaciones hidrometeorológicas, ya que a través de diversos estudios se pueden conocer elementos contaminantes en el aire, el agua y el suelo, con lo cual se afecta la seguridad alimentaria, la nutrición y se pueden propagar variedad de enfermedades. Según [30], la contaminación ambiental del aire, el agua y el suelo por productos químicos cancerígenos, causa entre el 1% y el 4% de todos los casos de cáncer. Por otro lado, las radiaciones ultravioleta, especialmente las producidas por el sol son perjudiciales para el ser humano, provocando diversos tipos de cáncer cutáneo, quemaduras de sol, envejecimiento acelerado de la piel, cataratas y otras enfermedades oculares[31]. En el año 2000 se diagnosticaron en el mundo más de 200.000 casos de melanoma y se produjeron 65.000 muertes asociadas a este tipo de cáncer. Otro ejemplo de ayuda en la salud se da en la zona Africana conocida como el "cinturón de la meningitis" [32], donde basados en los pronósticos de tormentas de viento

generan alertas tempranas a epidemias, permitiendo a diferentes tipo de organizaciones de salud, diseñar programas de vacunación de forma más eficaz.

#### **3.4.4 Producción alimentaria**

El beneficio de la observación de variables hidrometeorológicas, se ve reflejado con mayor profundidad en campos como la agricultura, la ganadería y la pesca donde los estudios medioambientales son útiles para la planificación de cultivos, producción de alimentos, información de crecientes o sequías de los niveles de agua, ya que al obtener información precisa de diferentes variables naturales, se llega a definir las estrategias de funcionamiento a largo plazo [25], con el fin de adaptar sus prácticas a las condiciones cambiantes, y garantizar la producción continua adaptada a las necesidades.

La generación de estudios con datos hidrometeorológicos es la base fundamental para hacer frente a la variabilidad y al cambio climático, analizar causas y generar propuestas en pro del mejoramiento del medio ambiente. Para todos los campos de la sociedad es importante tener un sistema de observación y registro de variables ambientales, ya que con el conocimiento de la climatología en general se pueden brindar soluciones a diversas problemáticas y mitigar las condiciones de riesgo que afectan la vida humana.

#### **3.5 Revisión del estado del arte**

Como base fundamental para el desarrollo del proyecto se realiza el análisis de antecedentes y diversas fuentes de información bibliográfica con el fin de definir la tecnología y elementos necesarios para el posterior diseño de los componentes del prototipo desarrollado.

Existen una serie de proyectos que han propuesto desarrollos similares al prototipo planteado para la medición de diferentes variables hidrometeorológicas, utilizando registradores de datos comerciales (dataloggers) como los utilizados en [33] y [34], donde además se desarrolla software para la visualización de información de forma gráfica, conectándose a computadores mediante protocolos de comunicación cableados. Así

mismo se utilizan tarjetas externas especializadas, como la que se utiliza en [35] donde adicionalmente implementan sensores para el monitoreo de variables meteorológicas.

Otros desarrollos similares basan su funcionamiento en dispositivos microcontrolados, implementando estaciones meteorológicas de bajo costo como se explica en [36] y [37], en los cuales se integran sensores comerciales de la línea estudiantil para la medición de variables, y donde además se utilizan sistemas de comunicación inalámbrica, como los mostrados en [38] y [39] donde por medio de enlaces de radio y celular, se transmite la información adquirida a locaciones remotas con el fin de almacenarla y analizarla posteriormente.

También se encuentran proyectos donde utilizan computadores de placa reducida como la Raspberry Pi, y diversas herramientas de hardware y software libre para el desarrollo de interfaces gráficas, como en [40] y [41], donde se realiza la adquisición y procesamiento de información dentro del sistema embebido. Así mismo trabajos como los mostrados en [42] y [43], utilizan las diferentes interfaces de la Raspberry Pi para la conexión de sensores hidrometeorológicos especializados y la integración de módems para el acceso a internet transmitiendo la información a servidores web y aplicaciones móviles en smartphones. De igual forma el proyecto presentado en [44], utiliza la Raspberry Pi para el monitoreo de variables meteorológicas integrando un sistema de adquisición de imágenes mediante una cámara digital.

Comercialmente existen sistemas de registro de información basados en estaciones meteorológicas, por lo cual, en la actualidad existen gran variedad de fabricantes de sensores y plataformas colectoras de datos. En el mercado colombiano se encuentran diferentes plataformas colectoras de datos (DCP) de fabricación extranjera, como la **NetDL1000 de OTT** [45], la **CR6 de Campbell Scientific** [46] y la **DA15K de Siap+Micros** [47], las cuales son de uso profesional y cuentan con especificaciones técnicas de calidad superior, para la adquisición de datos de sensores hidrometeorológicos especializados ya sean de tipo análogo, digital y diferentes protocolos de comunicación como RS232, RS485 y SDI-12. Además este tipo de plataformas puede integrar sistemas de transmisión de datos que puede ser de tipo radio, celular o satelital.

A continuación se presenta en resumen los fabricantes y las características técnicas de los equipos con los que actualmente cuenta la red de estaciones automáticas del IDEAM y además el valor aproximado en el mercado.

Sistema de Adquisición de datos: <b>OTT NetDL 100 y NetDL 500</b>	
<b>Fabricante</b> : OTT	<b>País</b> : Alemania
Empresa alemana con más de 140 años de experiencia formada por tres empresas (OTT, Hydrolab, y Adcon Telemetría) especializada en brindar soluciones con productos para la medición de agua, registro de datos y telemetría, soluciones de software, meteorología y sistemas integrados.	
<b>Especificaciones:</b> Display and Jog shuttle Communication interfaces: - USB Host/Device - 2 x RS-232 (full DB9) - LAN 10Mbit, 10BaseT Interface: - SDI-12 - RS-485 - 4 x Impulse/Frequency/Status - 2x Switching contact -4x analogue input - Standard + GSM Modem	
Valor: <b>\$15.000.000 COP</b>	

Tabla 4. Plataforma OTT NetDL1000 y NetDL500

Fuente: [www.ott.com](http://www.ott.com)

Sistema de Adquisición de datos: <b>SUTRON 8310</b>	
<b>Fabricante</b> : Sutron	<b>País</b> : Estados Unidos
Empresa americana especializada en sistemas Hidrológicos, Meteorológicos y Oceánicos, proporciona los sistemas y productos para la recolección de datos en tiempo real para dicha tecnología, dentro de sus productos se encuentran: Registradores y Controladores, módulos de I/O, Sensores, comunicaciones y telemetría, software de administración de datos.	
<b>Especificaciones:</b> - Analog Channels 8 - Input Voltage -0.1 to 5V -Digital Inputs 6 (2 Bi-Directional) - Digital Outputs 2 (Bi-Directional) - Communication Ports 4 - RS-232, RS-485, USB, 2 SDI-12 - LCD Display Type 2x20 - Keypad Type 6 buttons - SD Card Download data & read/write - Ethernet Built-in -GPRS Modem Built-in	
Valor: <b>\$11.000.000 COP</b>	

Tabla 5. Plataforma Sutron 8310

Fuente: [www.sutron.com](http://www.sutron.com)

Sistema de Adquisición de datos: <b>Unilog Com</b>	
<b>Fabricante</b> : SEBA	<b>País</b> : Alemania
Empresa alemana con más de 40 años de experiencia en la fabricación de modernos y avanzados instrumentos de control de medida para hidrología, meteorología y calidad de aguas.	
<b>Especificaciones:</b> - alimentación 5,5...20V. - memoria serial flash con 4Mbyte - display (3 líneas, 16 cifras) - teclado con 3 botones <b>Interfaces:</b> - RS232, RS485, USB, LAN <b>Entradas:</b> - RS 485 (SHWP) - SDI12 - arriba/abajo contador de fase, pulsos (lluvia) - 2 entrada de contractos - 8 entradas bi/Unipolares	
Valor: <b>\$14.000.000 COP</b>	

Tabla 6. Plataforma SEBA UnilogCom  
 Fuente: www. www.seba-hydrometrie.com

Sistema de Adquisición de datos: <b>DA15K</b>	
<b>Fabricante</b> : SIAP+MICROS	<b>País</b> : Italia
Empresa que nace en el 2003 de la fusión de SIAP Bologna (1925) y MICROS (1976), dos empresas líderes en el mercado de instrumentos de medición para el monitoreo ambiental.	
<b>Especificaciones:</b> Procesador: CPU ARM9 (32 bit RISC) 240Mhz Memoria: SDRAM 64MB FLASH 512MB Sistema Operativo: Embedded: Windows CE 6.0 Display: 3.5" 320x240px, color, touch-screen <b>Interfaz de Comunicación:</b> -2xUSB Host -1x Ethernet 10/100 Base T -2x SDI-12 -1x GSM Modem -4x RS232 / RS485 <b>Ingresos analógicos:</b> -Diferencial: -2.5 – 2.5V -Single ended: 0 – 2.5V, 0–20mA, 4–20mA -4x Pt100	
Valor: <b>\$22.000.000 COP</b>	

Tabla 7. Plataforma Siap+Micros DA15K  
 Fuente: www.siapmicros.com

### 3.6 Sensores hidrometeorológicos utilizados

Para realizar la verificación del funcionamiento del prototipo se utilizaron sensores especializados que realizan mediciones a variables como temperatura del aire, humedad relativa del aire, presión atmosférica, precipitación, velocidad del viento y dirección del viento, los cuales cuentan con salidas análogas, digitales y seriales SDI-12. Estos dispositivos son facilitados por el IDEAM y son los utilizados actualmente en las estaciones hidrometeorológicas del país.

A continuación se describen las características y funcionamiento de cada uno de los sensores utilizados.

#### 3.6.1 Sensor de temperatura y humedad del aire

El sensor ROTRONIC HygroClip HC2 es utilizado especialmente en aplicaciones meteorológicas de alta precisión para la medición de temperatura y humedad del aire ya que posee un alta exactitud ( $\pm 0,8\%$  Humedad Relativa y  $0,1^{\circ}\text{C}$  en Temperatura). Está compuesto por un sensor de tipo capacitivo para la medición de humedad relativa y un sensor tipo PT100 1/3 Clase B, para la medición de temperatura. El voltaje de alimentación está en el rango de 7 a 15VDC y cuenta con salidas análogas de voltaje independientes para cada una de sus variables. En la siguiente tabla se puede observar los rangos de medición de cada una de las variables del sensor.

Variable	Salida Análoga	Rango
Temperatura	0 ... 1V <sub>DC</sub>	-40 ... 60 °C
Humedad	0 ... 1V <sub>DC</sub>	0 ... 100 %

Tabla 8. Rangos de medición sensor de temperatura y humedad del aire

El sensor es de fácil montaje e instalación, y cuenta con una sonda de dos partes que le permite ser conectado a cualquier *datalogger* y ubicarlo a diferentes distancias, adicionalmente cuenta con una carcasa anticorrosión y anti radiación la cual protege al sensor de diversas condiciones meteorológicas aumentando la durabilidad, el rendimiento y la fiabilidad de las mediciones del sensor. En la Figura 10 se observa el sensor utilizado.



Figura 10. Sensor de temperatura y humedad del aire  
Fuente: Ficha técnica Rotronic HC2

### 3.6.2 Sensor de presión atmosférica

Para la medición de la presión atmosférica se utiliza el sensor VAISALA PTB100B, el cual en una pequeña carcasa de plástico contiene, como elemento sensible, un sensor capacitivo de silicio que ayudado por una válvula de entrada de aire, realiza la medición comparando la presión interna y externa. Está diseñado para realizar mediciones en interiores, a diferentes temperaturas, y es útil para aplicaciones aeronáuticas, industriales y medioambientales ya que tiene una exactitud de  $\pm 0.3$  hPa en el rango de operación. El sensor tiene una salida análoga que va de 0 a 5 V<sub>DC</sub> equivalente a 600 y 1060 hPa respectivamente. En la Figura 11, se puede observar el sensor de presión atmosférica.

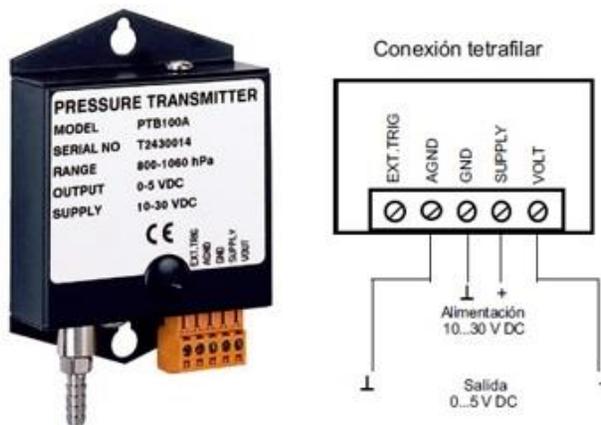


Figura 11. Sensor de presión atmosférica  
Fuente: Ficha técnica VAISALA PTB100B

### 3.6.3 Sensor de precipitación

Para la medición de precipitación se utiliza el sensor RIMCO 8500, el cual es un pluviómetro diseñado y construido para funcionar bajo diferentes condiciones climáticas, por un largo periodo con un mantenimiento mínimo. El principio de funcionamiento de este pluviómetro mostrado en la Figura 12, se conoce como de tipo balancín, la lluvia se recoge por medio de un colector en forma de embudo cónico, el cual hace pasar el agua a través de un orificio de precisión, donde cae en un balancín con 2 cazoletas, a medida que cada cazoleta se va llenando, el balancín se desequilibra lo cual hace que un imán active un *reedswitch* generando un pulso digital.

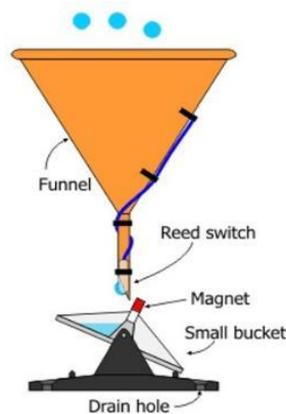


Figura 12. Pluviómetro tipo balancín

Fuente: <http://www.kifissiameteo.gr>

Este instrumento cuenta con todas las recomendaciones para la medición de lluvias, establecidas por la OMM, de acuerdo con las especificaciones técnicas del sensor cada pulso digital tiene una resolución equivalente a 0.1mm de agua<sup>1</sup>. Este pulso digital puede ser leído fácilmente por cualquier dispositivo de almacenamiento de datos basado en microcontroladores o microprocesadores. En la Figura 13, se puede observar el sensor utilizado.

---

<sup>1</sup> Según la OMM el milímetro (mm) -no mililitro (mL)- es la unidad como se mide la lluvia que llega al suelo, es igual a 1 litro por metro cuadrado.



Figura 13. Pluviómetro RIMCO 8500  
Fuente: Ficha técnica RIMCO 8500

### 3.6.4 Sensor de dirección y velocidad del viento

La dirección y velocidad del viento se mide con el sensor VAISALA WMT702, el cual es un anemómetro con 3 transductores separados  $120^\circ$  que realizan la medición de las variables utilizando el ultrasonido basado en la dirección y el tiempo que tarda en viajar de un transductor a otro. Este sensor posee una estructura completamente de acero resistente al agua, la contaminación y la corrosión, con certificado IP67. El sensor no tiene partes móviles lo cual hace que prácticamente no requiera mantenimiento periódico. En la Figura 14, se puede observar el sensor utilizado.



Figura 14. Sensor de dirección y velocidad del viento  
Fuente: Ficha técnica VAISALA WMT702

Este sensor es de alta fiabilidad ya que cuenta con especificaciones técnicas de alta calidad, cumple con los requisitos establecidos en la guía de recomendaciones de la OMM [11], lo cual hace que sea ideal para aplicaciones meteorológicas, adicionalmente se puede usar en aplicaciones aeronáuticas ya que cumple con los requisitos de la Organización de Aviación Civil Internacional (OACI). En la Tabla 9, se observan las especificaciones técnicas del sensor.

<b>VAISALA WMT 702</b>		
	<b>Velocidad del Viento</b>	<b>Dirección del viento</b>
Rango de Medición	0 ... 65 m/s	0 ... 360 °
Exactitud	±0.1 m/s o 2% de la lectura	±2 °
Umbral de Inicio	0.01 m/s	0.1 m/s
Resolución	0.01 m/s	0.01 °
Tiempo de Respuesta	250 ms	250 ms

Tabla 9. Especificaciones sensor de dirección y velocidad del viento

El sensor funciona con una fuente de alimentación de 9 a 36 V<sub>DC</sub> y cuenta con interfaces de comunicación RS485, RS422, RS232 y SDI-12, esta última es utilizada para la comunicación con el prototipo desarrollado, permitiendo realizar la medición de las variables de dirección y velocidad del viento utilizando este protocolo como se explicó en la sección 4.4.3. En la Figura 15, se identifican los comandos necesarios para la adquisición de los datos y el mensaje de respuesta, en el que se puede observar la respuesta: “a<WS><WD><x><y><s><CR><LF>” donde los datos de velocidad (<WS>) y dirección (<WD>) están ubicados en las posiciones 1 y 2 respectivamente.

### Mensaje para el comando C y M de WS425 A/B SDI-12

Cuando ingresa los comandos **aC!** o **aM!** seguidos de **aD0!**, la respuesta es la siguiente:

a<WS><WD><x><y><s><CR><LF>

donde

- a = La dirección de un dígito del WMT700 que corresponde al primer carácter del valor **address** (dirección)
- <WS> = Velocidad del viento polar en unidades seleccionadas  
Formato: +(ss)s.s
- <WD> = Dirección del viento polar en grados  
Formato: +(dd)d.d
- <x> = Componente x de la velocidad del viento en unidades seleccionadas  
Formato: ±(ss)s.s
- <y> = Componente y de la velocidad del viento en unidades seleccionadas  
Formato: ±(ss)s.s
- <s> = Velocidad del sonido estática en millas por hora  
Formato: ±(ss)s.s
- <CR><LF> = Termina la respuesta.

Figura 15. Respuesta SDI-12 sensor de dirección y velocidad del viento

### 3.7 Software de monitoreo remoto Hydras3

Como herramienta de monitoreo remoto se utiliza el software **Hydras3**, el cual es un programa licenciado desarrollado por la empresa alemana **OTT Hydrmet**, que es utilizado por el IDEAM para realizar análisis de datos hidrometeorológicos, presentación de informes y configuración de alertas.

El banco de datos de Hydras3 está instalado en un servidor central y cuenta con dos plataformas para el acceso a la información; el acceso mediante un software de escritorio y el acceso mediante una página web.

### 3.7.1 Versión de escritorio Hydras3

El software de escritorio se instala en cualquier computador personal y realiza la conexión al servidor central accediendo a la información únicamente por medio de una red de área local (LAN). Este software permite gran variedad de funcionalidades, tales como:

- Comunicación con instrumentos meteorológicos.
- Administración de estaciones.
- Administración de series de tiempo.
- Inclusión de mapas e ilustraciones.
- Evaluación gráfica.
- Superposición de gráfica múltiple.
- Editor gráfico.
- Representación numérica de datos medidos.
- Evaluación estadística.
- Creación de sensores virtuales.
- Análisis de correlación.
- Representación de isolíneas.
- Importación de valores medidos.
- Exportación de valores medidos.

En la Figura 16, se presenta una captura de pantalla de la ventana inicial de la versión de escritorio del software Hydras3, donde se puede ver al costado izquierdo el listado de estaciones y sensores, en la parte central la localización geográfica en el mapa de Colombia y al costado derecho la descripción de ubicación.

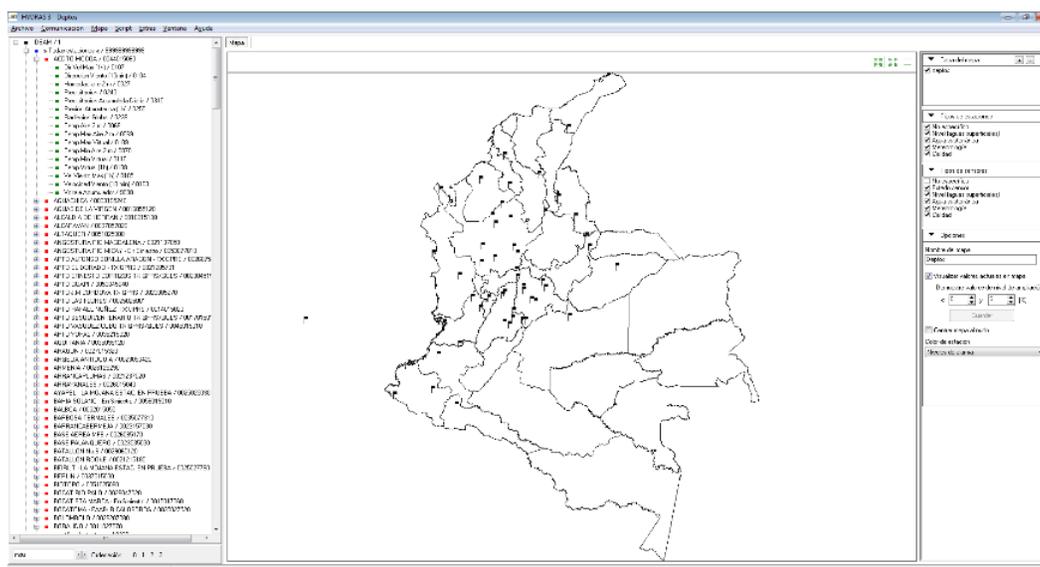


Figura 16. Interfaz de escritorio Hydras3

Fuente: IDEAM

Dentro de las funcionalidades más importantes del software esta la presentación de la información en forma gráfica y en forma tabular la cual sirve para realizar el análisis e interpretación de los valores de cada una de las variables hidrometeorológicas. En la Figura 17, se observa la representación gráfica y numérica de una variable en el software de escritorio Hydras3.

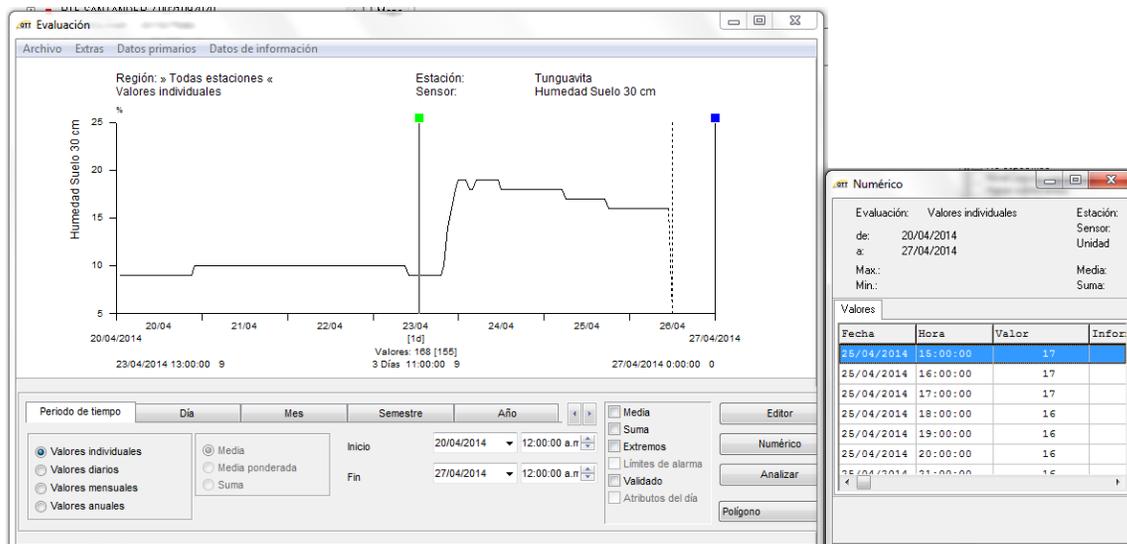


Figura 17. Presentación gráfica y tabular – versión de escritorio software Hydras3

Fuente: IDEAM

El software de escritorio Hydras3 permite también realizar la superposición de varios valores, a fin de poder hacer la comparación de dos o más variables, ya sean de sensores de una misma estación o de diferentes localizaciones, permitiendo a los usuarios interpretar comportamientos basados en representaciones gráficas y que además sean útiles para la presentación de informes. En la Figura 18, se puede observar la superposición de múltiples gráficas en el software de escritorio Hydras3.

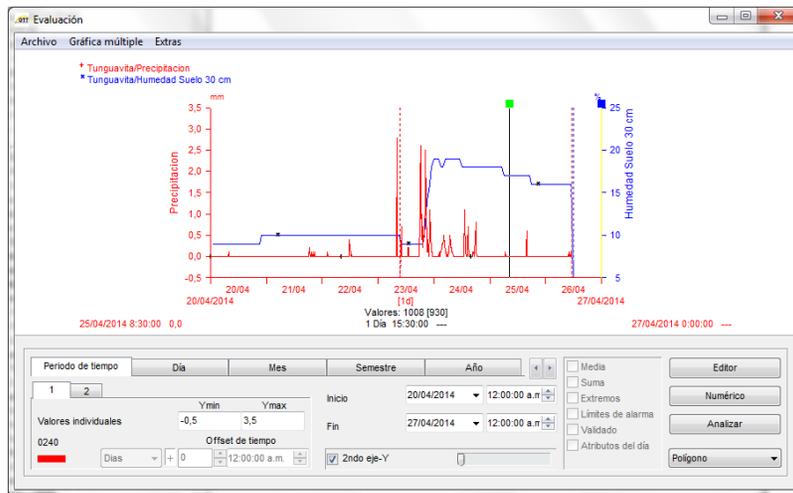


Figura 18. Superposición de múltiples gráficas en Hydras3

Fuente: IDEAM

Otra funcionalidad importante del software de escritorio es la configuración de valores límite para generación de alertas, lo cual permite identificar cuando un valor supera los niveles establecidos, los diferentes tipo de alertas se representan con un código de colores, los cuales generalmente son amarillo para nivel de riesgo bajo, naranja para nivel de riesgo medio y rojo para nivel de riesgo alto, de esta manera se establecen criterios para la generación de boletines y reportes que ayuden en a la prevención de diferentes fenómenos. En la Figura 19 se observa la configuración de alarmas en el software de escritorio Hydras3.

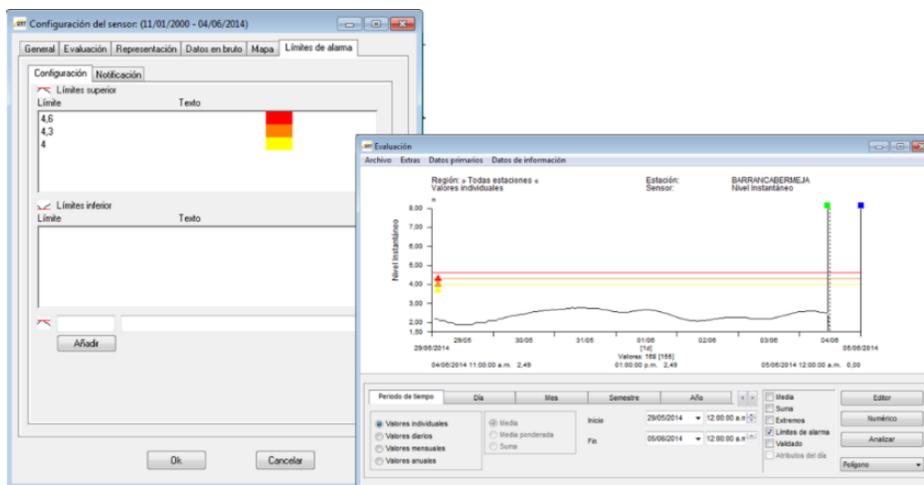


Figura 19. Configuración de límite de alarma Hydras3.

Fuente: IDEAM.

### 3.7.2 Versión WEB Hydras3

El acceso vía WEB a la plataforma Hydras3 se realiza mediante cualquier explorador a través de la página [hydras3.ideam.gov.co](http://hydras3.ideam.gov.co) la cual después de ingresar un nombre de usuario y contraseña autorizados, se tiene acceso a toda la información de las estaciones, el uso de la interfaz web tiene la ventaja de poder acceder desde cualquier parte del mundo con acceso a internet. En la Figura 20 se observa la interfaz web de Hydras3.

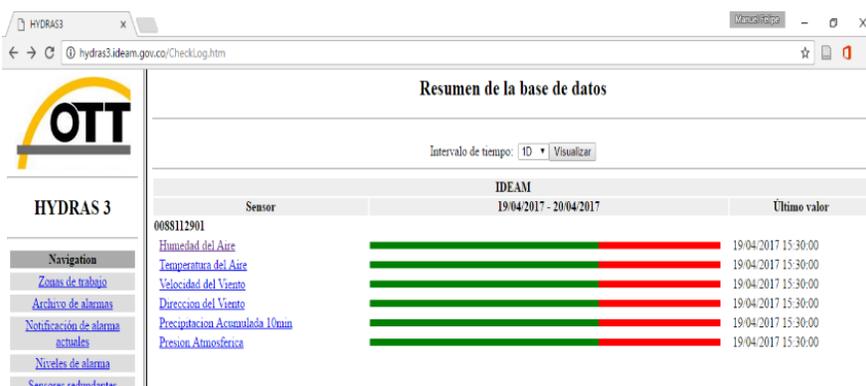


Figura 20. Interfaz WEB Hydras3.

Fuente: IDEAM.

A diferencia de la versión de escritorio, la versión web es más básica, limitando las funcionalidades de Hydras3 únicamente a la visualización de datos en forma gráfica y en forma tabular. En la Figura 21, se puede observar la representación gráfica de una variable en la interfaz web de Hydras3.

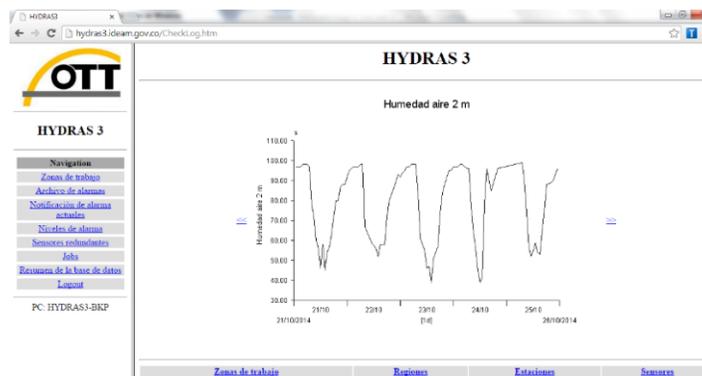


Figura 21. Representación gráfica interfaz WEB Hydras3.

Fuente: IDEAM.

## 4 DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

### 4.1 Plataforma de desarrollo utilizada (Raspberry Pi)

Debido a que uno de los objetivos específicos era utilizar un computador de placa reducida para la adquisición y procesamiento de datos, se descartan sistemas microcontrolados como los que maneja Atmel, Microchip o Freescale, por lo cual se hizo una revisión de algunas de las plataformas de desarrollo más vendidas comercialmente, y con mejores funcionalidades, reduciendo la selección a las tarjetas **Raspberry Pi** y **Beaglebone**, debido a su capacidad para manejar sistemas operativos con la posibilidad de utilizar diversas herramientas de software. En la Tabla 10, se muestran las características de los dispositivos tenidos en cuenta.

	Raspberry Pi	Beaglebone
<b>Modelo</b>	2 Model B	Black Rev C
<b>Precio</b>	COP\$ 175.000	COP\$ 290.200
<b>Procesador</b>	900Mhz Quad Core ARM	1Ghz ARM Processor
<b>RAM</b>	1 GB	512MB DDR3
<b>Almacenamiento</b>	Micro SD Card (Required) Expandable via USB	4 GB on board flash storage. Expandable by USB & Micro SD
<b>Audio</b>	Stereo over HDMI or 3.5mm Jack	Stereo over HDMI
<b>Video</b>	Full Size HDMI port, DSI port	Micro HDMI
<b>GPIO</b>	26/40 Pins	65/92 Pins
<b>Entradas Análogas</b>	0	7 (1.8V 12bit)
<b>Interfaces</b>	UART, Bus I2C, Bus SPI	UART, I2C, SPI, PWM, LCD, CAN
<b>Conector Cámara</b>	CSI port	No
<b>Periféricos</b>	4 USB ports	1 USB port
	1 10/100 Ethernet port	1 10/100 Ethernet port
<b>Consumo</b>	800mA	250mA
<b>Fuente de Alimentación</b>	Micro USB	Micro USB or 5VDC connection

Tabla 10. Comparación Raspberry Pi 2 Model B y Beaglebone Black Rev C

Para el desarrollo del prototipo se decidió utilizar la tarjeta **Raspberry Pi 2 Model B** [48], la cual es un computador de placa reducida, que cuenta con las características técnicas

suficientes para el desarrollo de la aplicación. A pesar que la **Beaglebone** cuenta con un conversor análogo digital a 12bit interno, el voltaje máximo que esta maneja es de 1.8V, lo cual impediría integrar sensores que manejan voltajes superiores (2.5V y 5V), la **Raspberry Pi 2** tiene un menor costo, y cuenta con el doble de memoria RAM (1GB), además posee 4 puertos USB, 1 puerto Ethernet, 1 puerto dedicado para la conexión de cámara (CSI), 1 puerto especial para adherir una pantalla *touch* (DSI), los cuales son accesorios oficiales que se consiguen de en el mercado, facilitando el desarrollo de los objetivos del proyecto, así mismo cuenta con un conector de 40 pines, los cuales permiten la interacción con otros dispositivos de lógica digital, cada uno de los 40 pines tiene una función específica, ya sea para alimentación (3.3V, 5V, GND), protocolos de comunicación (SPI, I<sup>2</sup>C, UART), o como puertos digitales configurables de entrada/salida a 3.3V (pines GPIO – *General Purpose Input Output*). En la Figura 22, se puede observar la placa Raspberry Pi 2 Model B y la distribución y uso de puertos e interfaces utilizados en el desarrollo del prototipo.

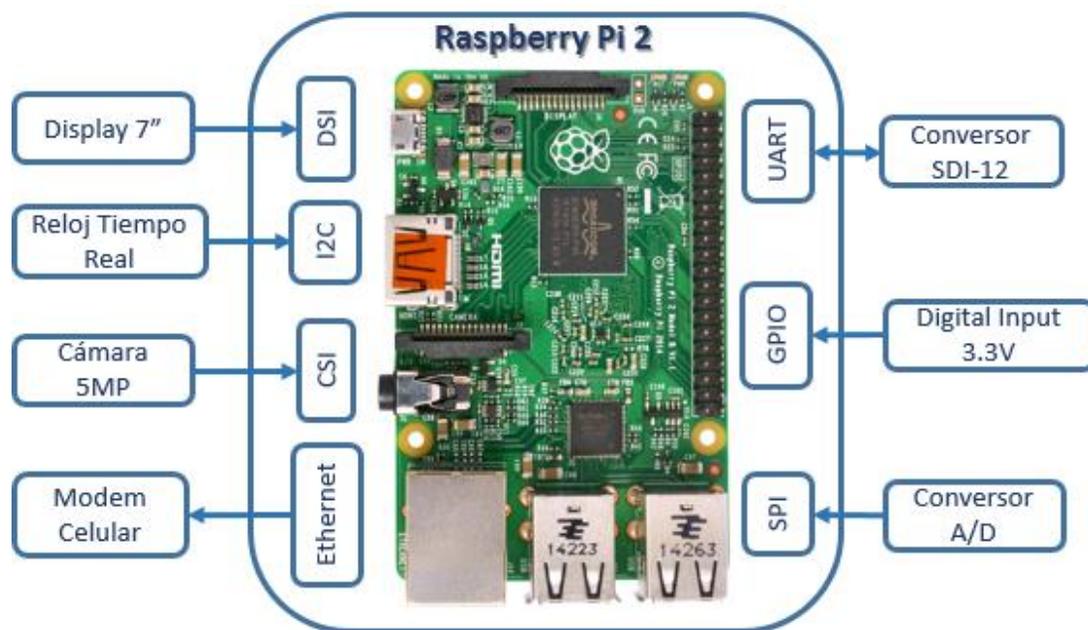


Figura 22. Distribución de puertos e interfaces en la tarjeta Raspberry Pi

Fotografía Raspberry Pi tomada de: [www.raspberrypi.org](http://www.raspberrypi.org)

Para este proyecto se utiliza el sistema operativo, **Raspbian Jessie** en su versión del **18-03-2016**, la cual es una distribución *open-source* adaptada de Linux Debian, que es

conocida como el sistema operativo oficial de Raspberry. Este sistema operativo cuenta con una interfaz gráfica “amigable” que permite instalar y ejecutar programas funcionales, de manera similar a otros sistemas basados en Linux. En la Figura 23 se presenta una captura de pantalla de la interfaz gráfica del sistema operativo utilizado.

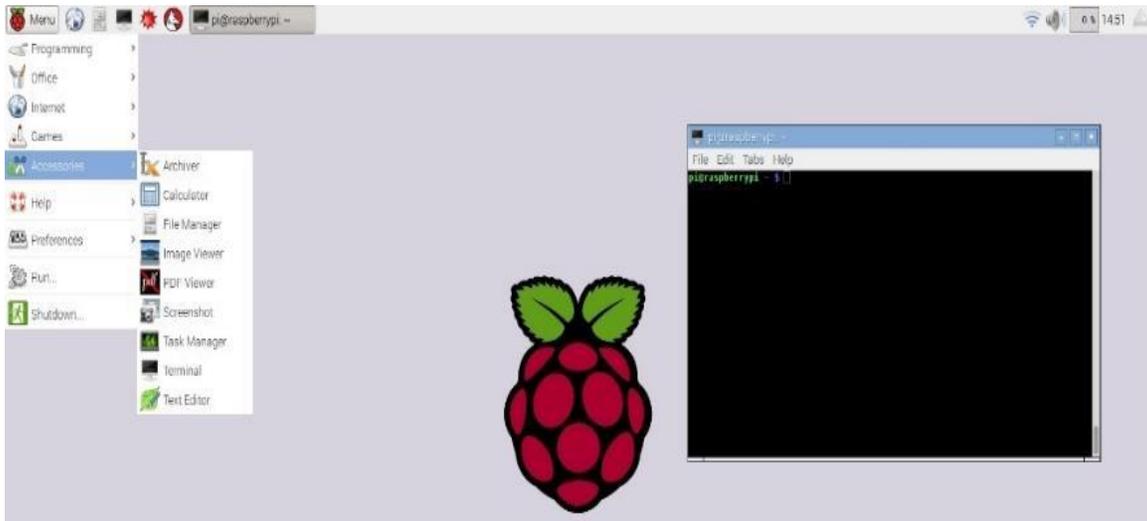


Figura 23. Captura de pantalla del sistema operativo Raspbian Jessie

## 4.2 Configuración del reloj en tiempo real

A pesar de que la Raspberry Pi puede actualizar la zona horaria (fecha y hora) mediante la sincronización a través de internet por medio de un servidor NTP (*Network Time Protocol*), se implementó un sistema de respaldo adicional, debido a la necesidad de registrar la fecha y hora real en la que se adquiere la información de los sensores. Por lo tanto se añade al sistema un reloj de tiempo real externo (*Real Time Clock - RTC*), el cual es un circuito integrado que permite mantener la fecha y la hora actualizadas cada vez que se inicie el sistema, ya sea por alguna falla en el sistema de alimentación o simplemente por el apagado del dispositivo. Para esta tarea se utiliza el dispositivo DS1307, el cual es un circuito integrado que se consigue fácilmente en el mercado y que se caracteriza por su sencillez de uso y confiabilidad a largo plazo, adicionalmente brinda la posibilidad de conectar una pequeña batería externa para que el reloj siga funcionando cuando el sistema se encuentre desconectado de la alimentación principal. En la Figura 24, se puede observar el módulo utilizado que incluye el integrado DS1307.



Figura 24. Módulo RTC - DS1307

La conexión entre la tarjeta Raspberry Pi y el módulo DS1307 se realiza por medio del protocolo I<sup>2</sup>C, el cual es un protocolo de comunicación serial síncrono de baja velocidad, que funciona en configuración maestro esclavo. Hay que tener en cuenta que para esta aplicación el módulo DS1307 funciona como esclavo, la alimentación (5V y GND) se toma de la fuente de alimentación de la tarjeta principal y se realiza la conexión directa entre los pines SCL (*System Clock*) y SDA (*System Data*) para la comunicación de datos. En la Figura 25, se muestra la conexión entre la Raspberry Pi y el módulo DS1307.

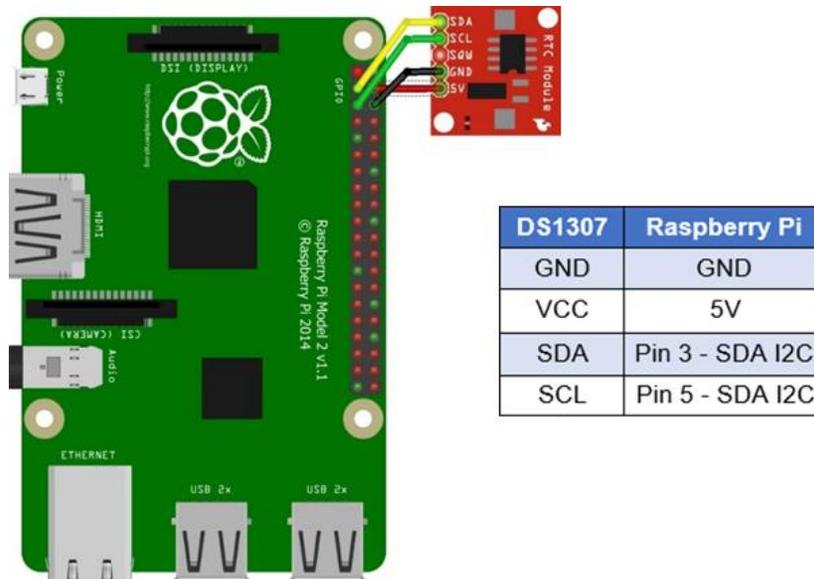
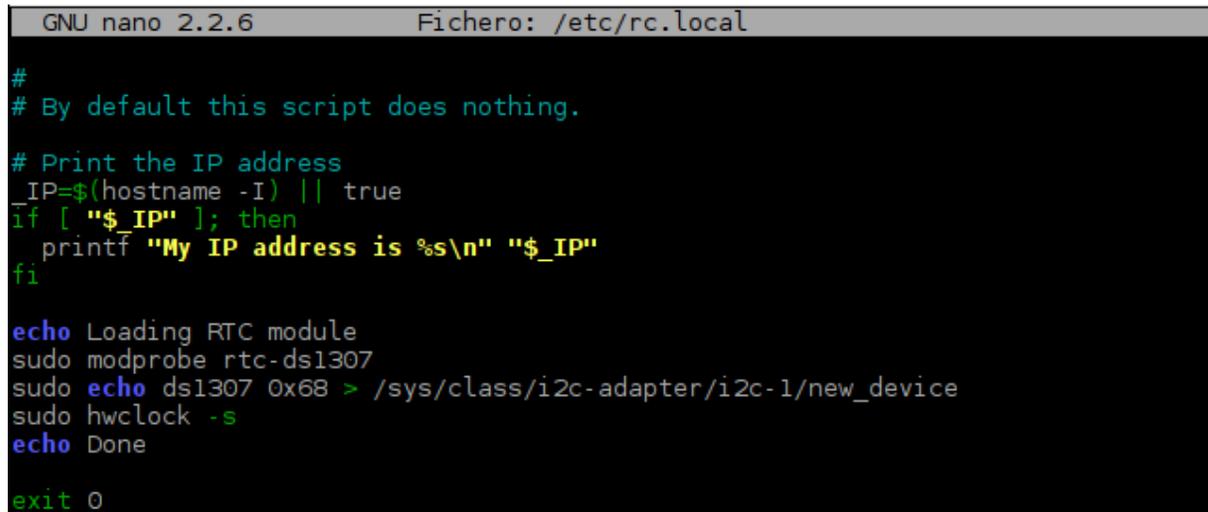


Figura 25. Conexión Raspberry Pi – Módulo DS1307

Una vez realizada la conexión, y con la fecha y hora actualizadas manualmente, se realiza una modificación al archivo ubicado en **etc/rc.local**, el cual se ejecuta en la secuencia de arranque de la Raspberry Pi, con lo cual se garantiza que cada vez que se inicia el sistema operativo se realiza la lectura de la fecha y hora desde el reloj de tiempo real. En

la Figura 26, se puede observar el fichero **rc.local** con la secuencia de inicio para la lectura del reloj de tiempo real. En el ANEXO A, se describe en detalle la configuración del reloj en tiempo real.



```
GNU nano 2.2.6 Fichero: /etc/rc.local
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
echo Loading RTC module
sudo modprobe rtc-ds1307
sudo echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
echo Done
exit 0
```

Figura 26. Secuencia de inicio reloj de tiempo real

### 4.3 Adquisición y procesamiento de datos

Para la adquisición y procesamiento de datos, se instaló y desarrolló el software necesario para el funcionamiento del prototipo. Inicialmente se realizó la instalación y configuración del motor de base de datos MySQL, en el que realizará el almacenamiento y organización de información mediante tablas de datos ordenadas como se explica en la sección 4.5.

Posteriormente se desarrolló una aplicación base utilizando Python3 la cual basada en la configuración establecida, utiliza los pines de propósito general de la Raspberry Pi para interactuar con dispositivos externos, encargándose de funciones como la lectura de datos de los sensores, su respectivo procesamiento y transformación en variables hidrometeorológicas, así como también la escritura de dicha información en las tablas de la base de datos. Esta aplicación en Python3 se encarga también de la creación del archivo de comunicación para el envío de datos y la conexión con el servidor remoto. Adicionalmente es la responsable de la adquisición y almacenamiento de imágenes

fotográficas. En las secciones 4.4, 4.6 y 4.8 se presentan los usos y funcionamiento de la aplicación base.

Adicionalmente utilizando el entorno de desarrollo integrado MonoDevelop y su compilador de C#, se desarrolló una interfaz gráfica, con el propósito de brindar al usuario un entorno amigable de configuración y visualización de cada una de las variables a medir, generando un archivo de texto plano en formato XML, en el que la aplicación de Python3 se basa para iniciar su funcionamiento. Esta interfaz muestra los valores adquiridos por los sensores, visualizándolos de forma tabular y de forma gráfica, de la misma manera admite configuración de la captura de imágenes, y la gestión sobre la base de datos permitiendo realizar consultas, exportaciones y borrado de la información. En la sección 4.7 se describe en detalle los componentes de la interfaz gráfica.

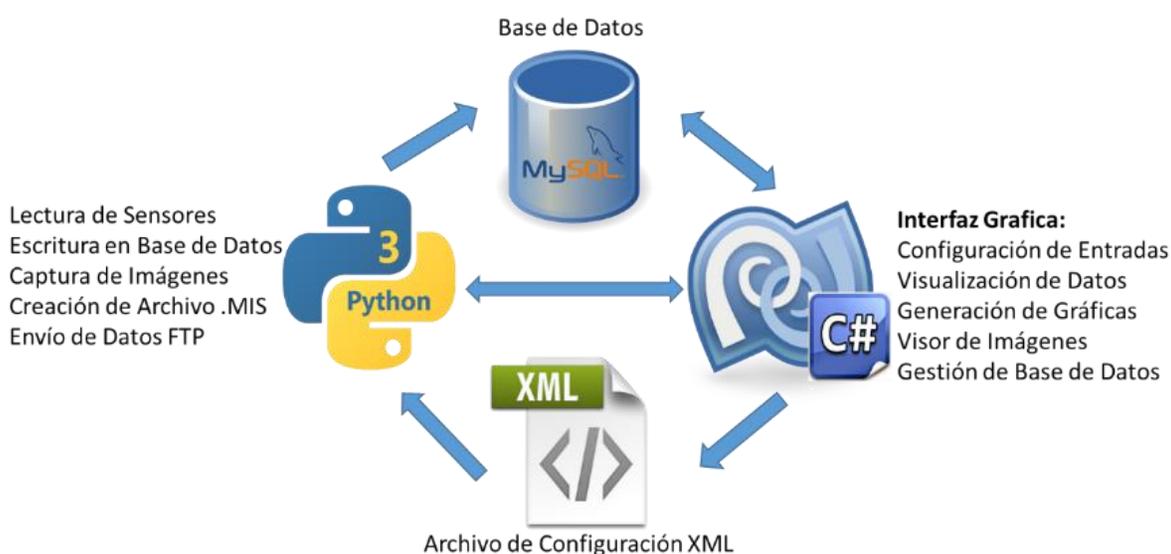


Figura 27. Estructura y componentes de software implementados en el prototipo.

Todos los componentes de software son de uso libre para cualquier persona y se comunican entre sí para el funcionamiento adecuado del prototipo implementado. En la Figura 27, se puede observar la estructura y componentes de software utilizados para el desarrollo del proyecto.

#### 4.4 Medición de variables hidrometeorológicas

Basados en la revisión de tecnologías existentes en la actualidad, y en la revisión de especificaciones técnicas de los sensores electrónicos más utilizados en el campo de la meteorología, se decidió implementar el prototipo de una plataforma colectora de datos con las siguientes especificaciones técnicas:

- ✓ Ocho (8) entradas de tipo análogo configurables por software, los cuales permitirán conectar sensores con salida de voltaje en los rangos de 0-1 V, 0-2.5 V, y 0-5 V.
- ✓ Dos (2) entradas digitales, para sensores con salida de pulsos.
- ✓ Una (1) entrada serial con protocolo SDI-12.

A continuación se describe cada una de las entradas utilizadas para la lectura de sensores.

##### 4.4.1 Entradas análogas

Debido a que la tarjeta Raspberry Pi no cuenta con pines para lectura de señales análogas, para la medición de sensores de este tipo, se utilizó el convertor análogo digital MCP3008, el cual es un dispositivo de bajo costo que cuenta con 8 canales de entrada a una resolución de 10 bits y una interfaz serial SPI (*Serial Peripheral Interface*), que lo hace compatible con otros dispositivos que manejen el mismo estándar, y en este caso particular compatible con la Raspberry Pi utilizada. Se decidió utilizar este dispositivo ya que de acuerdo con [11] las plataformas colectoras de datos normalmente cuentan con un número de canales análogos comprendido entre 4 y 32, los cuales son particularmente importantes puesto que la mayoría de transductores meteorológicos utilizados para el sensado de variables como temperatura, humedad y presión, entregan una señal de voltaje directo, proporcional a la medición. Por ejemplo, en [15], se hace referencia a diferentes tipos de sensores con sensibilidades de salidas análogas las cuales pueden variar de acuerdo a los fabricantes. En la Tabla 11, se muestra en resumen las sensibilidades típicas de varios sensores meteorológicos.

Sensor	Sensibilidad
Temperatura del Aire	55 mV/°C
Humedad del Aire	10 mV/%
Temperatura del Suelo	55 mV/°C
Velocidad del Viento	124.9 mV/ms <sup>-1</sup>
Dirección del Viento	13.9 mV/deg

Tabla 11. Valores de sensibilidad de diferentes tipos de sensores.

Con un conversor análogo digital a 10 bits como el MCP3008, y utilizando voltajes de referencia de 0 a 5V se obtiene una resolución de 4.88mV/bit, lo cual es suficiente para cubrir los cambios de acuerdo a la sensibilidad en la gran mayoría de sensores hidrometeorológicos que se encuentran en el mercado. En la Figura 28, se puede observar el dispositivo MCP3008. En el ANEXO B, se encuentra en detalle la configuración y puesta en funcionamiento del conversor.

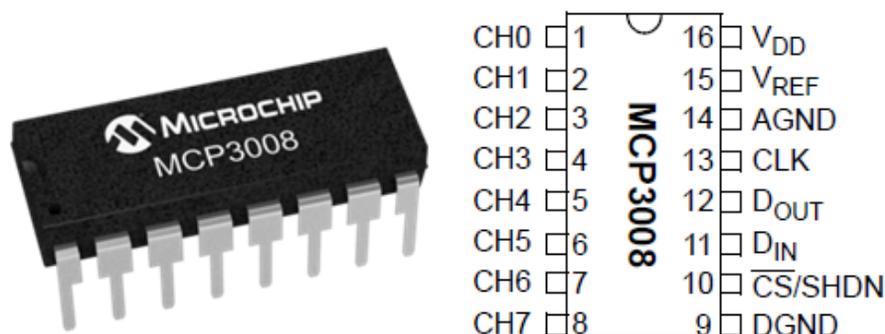


Figura 28. Conversor análogo digital MCP3008

Fuente: Hoja técnica MCP3008

Con el fin de realizar la conexión entre la tarjeta principal y el conversor análogo digital, se decidió utilizar un conversor de niveles lógicos basado en el transistor BSS138, ya que la Raspberry Pi maneja niveles lógicos de 0 V a 3.3 V mientras que el MCP3008 utiliza para la comunicación SPI niveles de 0 V a 5 V. Al utilizar el conversor de niveles lógicos se garantiza el correcto funcionamiento y la protección para los dos dispositivos. En la Figura 29, se observa el conversor de niveles lógico utilizado en el prototipo.

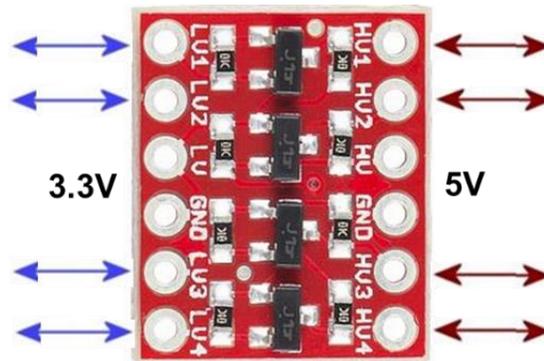


Figura 29. Conversor de niveles lógicos BSS138  
Fuente: Hoja técnica BSS138

En la Figura 30 se muestra la conexión entre la tarjeta Raspberry Pi, el conversor de niveles lógicos BSS138 y el conversor análogo digital MCP3008, conexión implementada en la construcción del prototipo.

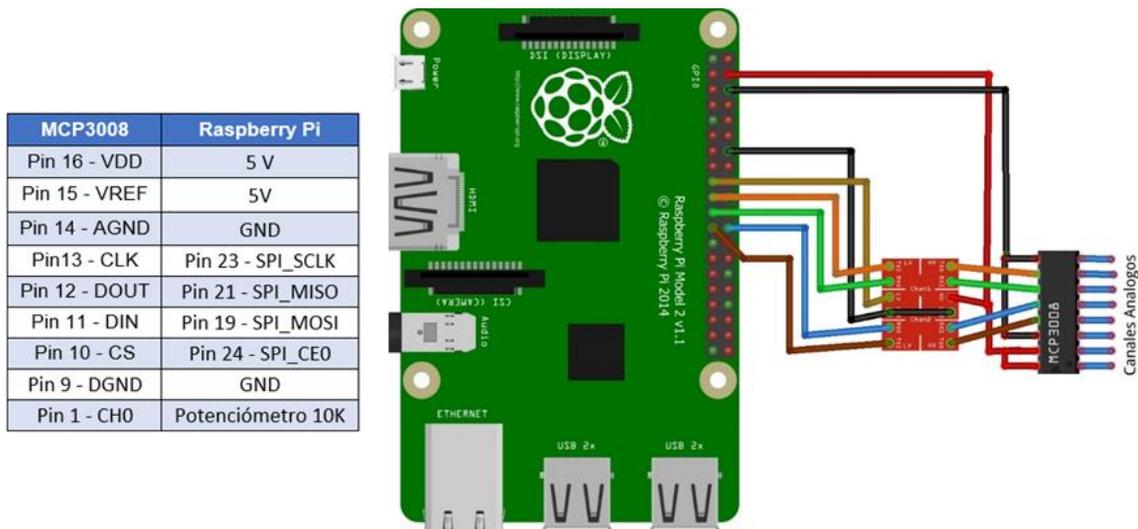


Figura 30. Conexión Raspberry Pi – BSS138 – MCP3008

El código base en Python3 realiza la lectura de la información entregada por el conversor análogo digital a través del protocolo SPI en formato digital, este dato se parametriza de acuerdo a los voltajes de referencia que en este caso son 0V y 5V, para establecer el valor de entrada medido en voltios.

Para determinar el valor del dato adquirido por un sensor con salida análoga de voltaje en las unidades correspondientes a la variable hidrometeorológica, en Python3 se crea la función **analogmap**, la cual realiza una linealización de los datos, transformando el valor de voltaje al valor correspondiente a la variable. Conociendo los valores de entrada y salida del sensor, se puede determinar el dato de salida basados en la Ecuación 1 mostrada a continuación:

$$Dato\ Salida = VoltajeEntrada \left( \frac{OutMax - OutMin}{InMax - InMin} \right) + OutMin \quad (1)$$

Donde:

**OutMin**=Dato Mínimo de Salida de la Variable.

**OutMax**=Dato Máximo de Salida de la Variable.

**InMin** = Dato Mínimo de Entrada en Voltios

**InMax** = Dato Máximo de Entrada en Voltios

La función **analogmap** solicita la información asociada al número del canal a medir, los datos máximos y mínimos de entrada y salida del sensor y el número de dígitos decimales con que se desea visualizar la variable, realiza el procedimiento correspondiente y retorna el valor calculado redondeado de acuerdo a las cifras decimales, es importante mencionar que todos los parámetros requeridos son ingresados por el usuario en la interfaz gráfica de configuración. En la Figura 31, se observa el código de funcionamiento de la función **analogmap**, y en el ANEXO F, se puede encontrar la librería completa que contiene las funciones utilizadas en la aplicación.

```
def analogmap(channel, Inmin, Inmax, Outmin, Outmax, decimal):
    volts = readSPI(channel)
    if volts>Inmax:
        volts=Inmax
    calculo=(volts*((Outmax-Outmin)/(Inmax-Inmin)))+Outmin
    out=round(calculo,decimal)
    return out
```

Figura 31. Definición de la función analogmap para la conversión voltaje a variable

#### 4.4.2 Entradas digitales

Los pines GPIO de la Raspberry Pi son configurables como entrada o salida digital a máximo 3.3V, además tienen la posibilidad de activar por software resistencias internas de *Pull-up* o *Pull-down* de acuerdo a la necesidad. Para la lectura de sensores con salida digital, cuya señal de salida está asociada a determinada cantidad de pulsos digitales, como los sensores de precipitación tipo balancín, se configuran dos puertos GPIO como entrada digital cada uno de ellos con una resistencia interna de *Pull-up* para garantizar la lectura del nivel alto y evitar ruidos en la señal, así mismo se utiliza una resistencia externa de 330Ω en cada pin, como protección en la lectura del pulso en nivel bajo. En la Figura 32 se muestra la conexión utilizada para las entradas digitales del prototipo.

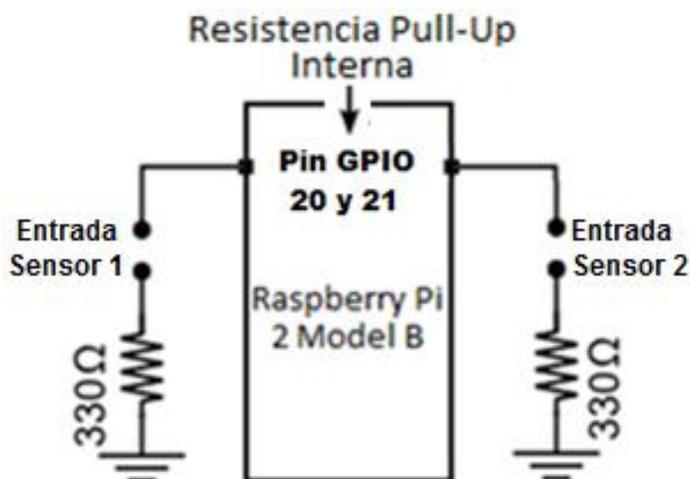


Figura 32. Conexión GPIO entradas digitales.

En el programa base desarrollado en Python3, para realizar la lectura de información de este tipo de sensores se hace uso de las interrupciones de programa, para este caso se utiliza la interrupción por flanco de bajada (**falling**), con la cual, cuando se detecta un cambio de estado en el pin, generado por un pulso digital, el programa principal se detiene para dar paso a una función predefinida, la cual, basada en un factor ingresado por el usuario de acuerdo a las especificaciones técnicas del sensor (resolución), realiza la suma de la cantidad de pulsos detectados multiplicados por el valor de la resolución. Por ejemplo, el sensor de precipitación **RIMCO 8500** descrito en la sección 2.6.3, cuenta con una resolución de 0.1mm por pulso, cuando el sensor genera 10 pulsos en la entrada

digital, la función realiza la conversión indicando un dato equivalente a 1mm. Posteriormente, al retornar de la interrupción, el programa continúa con la ejecución normal. En la Figura 33, se observa el código de funcionamiento de una interrupción en Python3, y en el **ANEXO F**, se puede encontrar el código completo de la aplicación.

```
def cuentaCH8(x):  
    global contadorCH8  
    time.sleep(0.5)  
    contadorCH8 = contadorCH8 + ResCH8  
    contadorCH8=round(contadorCH8,int(DecimalesCH8))  
  
GPIO.add_event_detect(pinCH8, GPIO.FALLING, callback=cuentaCH8)
```

Función llamada tras la interrupción

Declaración Interrupción

Figura 33. Funcionamiento de interrupción.

#### 4.4.3 Entrada serial SDI-12

Con la finalidad de poder integrar instrumentos de medición de última tecnología, se hizo uso del protocolo serial SDI-12 (*Serial Data Interface a 1200 Baud*), el cual es un estándar de comunicación asíncrona, que fue desarrollado para sensores e instrumentos inteligentes usados generalmente en el monitoreo de información medioambiental. SDI-12 utiliza un sistema Maestro-Esclavo, en la cual una unidad central de procesamiento, actúa como maestro y los sensores actúan como esclavos. Cada uno de los sensores tiene asociada una dirección de esclavo alfanumérico de tamaño igual a un carácter y que debe ser única dentro de cada bus de datos SDI-12, esta funcionalidad permite conectar hasta 62 sensores en paralelo, con direcciones de esclavos que van desde “0” hasta “9”, de “a” hasta “z” y de “A” hasta “Z”.

En la Figura 34, se muestra el ejemplo de la interconexión de un maestro (*data recorder*) con dos esclavos (sensores) utilizando el protocolo SDI-12, donde se puede observar una de las principales ventajas de este protocolo, ya que únicamente utiliza para su comunicación entre dispositivos un bus de datos de tres líneas, una línea para los datos seriales, una línea de tierra y una línea de voltaje de alimentación (*Serial data line, Ground line, 12-volt line*)

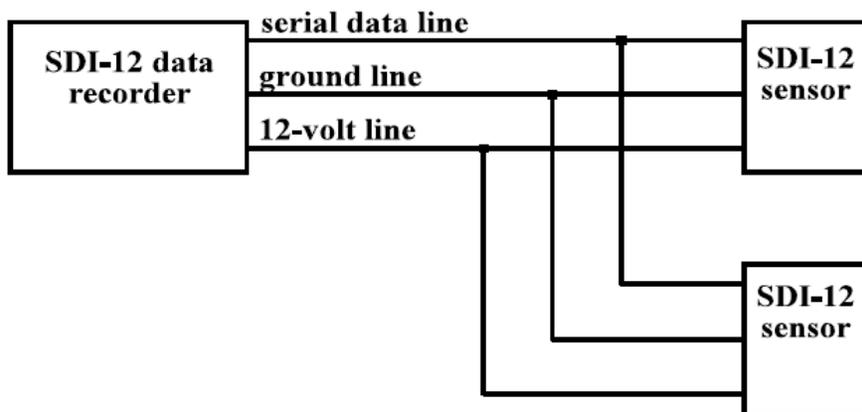


Figura 34. Ejemplo de interconexión del bus SDI-12

Fuente: Protocolo SDI-12 Versión 1.4

El protocolo SDI-12 se basa en el intercambio de caracteres ASCII entre el maestro quien envía una serie de comandos y el (los) dispositivo(s) esclavo(s) que genera respuestas a los comandos recibidos. En todos los comandos y respuestas el primer carácter siempre corresponde a la dirección de esclavo del dispositivo (en la nomenclatura del protocolo referenciado con la letra “a”) el último carácter de un comando siempre es el signo de exclamación “!”, el cual solo puede estar presente como finalizador. Los últimos dos bytes de una respuesta son un retorno de carro y avance de línea (<CR> <LF>) [49]. En la Figura 35 se observan algunas instrucciones del conjunto básico de comandos y respuestas utilizados en el protocolo SDI-12.

Name	Command	Response
Break	Continuous spacing for at least 12 milliseconds	None
Acknowledge Active	a!	a<CR><LF>
Send Identification	a!	allccccccmmmmmmvvvxxx...xx<CR><LF>
Change Address	aAb!	b<CR><LF> (support for this command is required only if the sensor supports software changeable addresses)
Address Query	?!	a<CR><LF>
Start Measurement*	aM!	atttn<CR><LF>
Start Measurement and Request CRC*	aMC!	atttn<CR><LF>
Send Data	aD0!	a<values><CR><LF> or a<values><CRC><CR><LF>

Figura 35. Conjunto básico de comandos/respuestas SDI-12

Fuente: [49]

En la Tabla 12, se observan las convenciones de los caracteres más utilizados en los comandos y respuestas del protocolo SDI-12.

Key	Meaning
a	Address
l	SDI-12 version number
c	8-character vendor identification
m	6-characters indicating sensor model number
n	Number of data values being returned
t	Time in seconds until data will be ready
v	3-character sensor version number
<value(s)> p.d.d	P – polarity sign
	d- numeric digits before the decimal place
	. – decimal point (optional)
	d – numeric digits after the decimal point
<CR><LF>	Terminates the response

Tabla 12. Convenciones de respuesta a comandos SDI-12

Fuente:[50]

Con el fin de implementar una entrada serial SDI-12 en el prototipo, como primera opción se evaluó la posibilidad de utilizar un conversor de protocolo de manera externa, los cuales se consiguen en el mercado de diferentes marcas y con posibilidad de conexión vía USB o RS232, como los mostrados en la Figura 36, Sin embargo este tipo de dispositivos se descartan por su alto costo (entre 100 USD y 200 USD) y además porque utilizan software de código cerrado para su funcionamiento, lo cual hace que la integración con tarjetas de desarrollo como Raspberry Pi sea de mayor complejidad.



Figura 36. Diferentes tipos de conversores SDI-12

Fuente: Internet

Como segunda opción, después de efectuar una revisión bibliográfica, se analizó el trabajo de grado realizado por la **USQ** (University of Southern Queensland) [50], en el cual realizan el desarrollo de una sistema de captura de datos para sensores SDI-12 basada

en Raspberry Pi, en este trabajo realizan la adecuación de software para la lectura de este tipo de sensores utilizando Python y para el control directo de los pines GPIO de la Raspberry Pi. Sin embargo, este tipo de lecturas requiere de un hardware externo compuesto por varios elementos electrónicos para la adecuación de la interfaz, de voltajes y de tiempos de conexión establecidos en protocolo SDI-12. En la Figura 37, se muestra el esquema de conexión utilizado por la USQ para la conexión de sensores SDI-12 a la tarjeta Raspberry Pi.

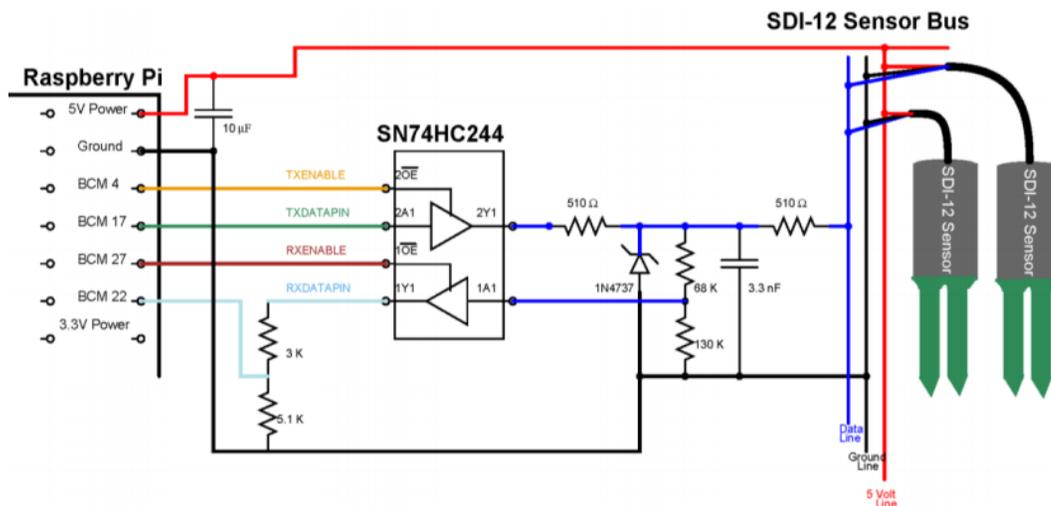


Figura 37. Ejemplo de conexión de sensores SDI-12 a Raspberry Pi.

Fuente: [50]

En el trabajo realizado por la USQ hacen referencia a la librería **Arduino-SDI-12**<sup>2</sup>, la cual se especializa en la lectura de sensores SDI-12 utilizando la placa Arduino (una plataforma electrónica abierta para la creación de prototipos basada en software y hardware libres), esta librería se encarga de la sincronización de tiempos y el intercambio de comandos conectando los sensores con salida SDI-12 directamente a los pines digitales de la placa Arduino sin depender de ningún elemento externo.

<sup>2</sup> Librería Arduino-SDI-12 originalmente escrita por Kevin Smith disponible en <https://github.com/StroudCenter/Arduino-SDI-12>

Debido a que la tarjeta Raspberry Pi ejecuta varios procesos para el funcionamiento adecuado del sistema operativo, así como de las funciones propias del prototipo, se decidió que para la lectura de sensores SDI-12 es preferible utilizar un sistema microcontrolado dedicado e independiente, que se encargue del intercambio de comandos realizando la lectura del dato y entregándolo directamente a la tarjeta Raspberry Pi. Por consiguiente, se decide utilizar la placa **Arduino Nano** para implementar la entrada SDI-12 en el prototipo, ya que este dispositivo es de bajo costo, tamaño pequeño, y permite la conexión física a la Raspberry Pi medio de un cable USB, con cual se efectúa la alimentación y la comunicación serial. En la Figura 38, se puede observar la conexión entre la tarjeta Raspberry Pi y el Arduino Nano.

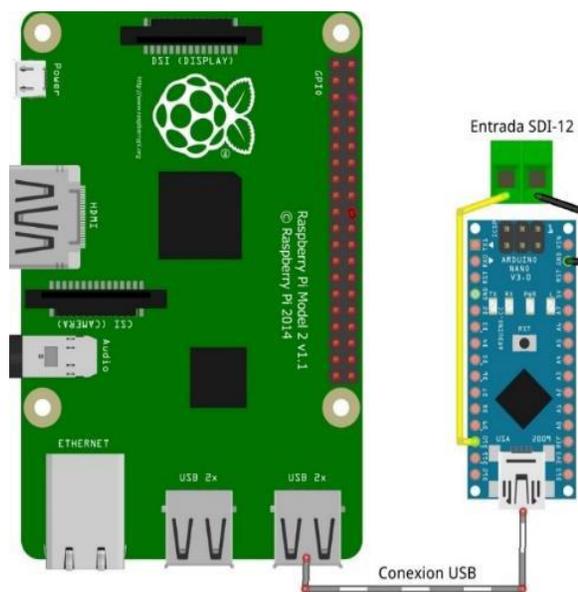


Figura 38. Conexión Raspberry Pi – Arduino Nano

#### 4.5 Almacenamiento de información en base de datos

Los datos de las mediciones realizadas por los sensores capturados por la aplicación base, se almacenan en una base de datos relacional, para lo cual se utilizó el motor MySQL, debido a que opera bajo software libre y cuenta con amplias características, robustez y facilidad de manejo, además tiene la capacidad de guardar información de forma ordenada y permite a los usuarios acceder a la información recopilada a través de tablas. Este sistema permite realizar consultas, actualizar, o modificar, los diferentes

registros de información, adicionalmente cuenta con interfaces de programación de aplicaciones (API's) y conectores disponibles para realizar la integración con Python3 y C#.

Se crean dos tablas para la visualización y almacenamiento de datos, en la tabla “**variables**”, se relacionan los **nombres, unidad, id y dato** de las variables hidrometeorológicas configuradas para la medición. La información de esta tabla se muestra en la ventana principal de la interfaz gráfica y el valor de cada una de las variables se actualiza de acuerdo al tiempo de visualización preestablecido por el usuario, con el fin de mostrar la información de los sensores en tiempo real.

En la tabla “**datos**”, se almacena el registro histórico de la información de las mediciones configuradas por el usuario, en esta tabla se relaciona el **id de la estación**, que sirve como identificador del sitio donde se realiza las mediciones, **id del sensor**, que sirve para la identificación del parámetro medido por la variable dentro del servidor remoto del IDEAM, la **Fecha y Hora** de ocurrencia en el momento en que se toma la medida del sensor, y el **dato**, o valor medido de la variable, con esta información almacenada el usuario podrá realizar diferentes tipos de análisis de acuerdo a las variables hidrometeorológicas medidas. En la Figura 39, se observa la estructura de las tablas creadas en la base de datos. En el ANEXO C, se encuentra en detalle la instalación y configuración básica del motor de base de datos.

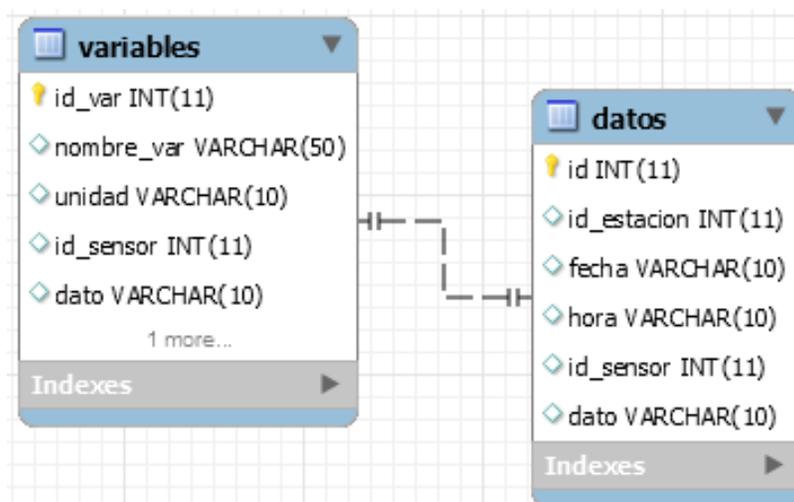


Figura 39. Tablas implementadas en la base de datos

## 4.6 Adquisición de imágenes

El propósito de la adquisición de imágenes, es realizar la captura de fotografías del entorno donde se encuentre ubicado el prototipo, de acuerdo a una condición periódica preestablecida por el usuario, guardando dichas imágenes dentro del espacio de almacenamiento de la Raspberry Pi, brindando la posibilidad de registrar gráficamente observaciones de fenómenos atmosféricos como meteoros<sup>3</sup> y nubes, cambios ambientales como el deshielo o variables hidrológicas como niveles de ríos y embalses, para así tener otra fuente de información con el fin establecer criterios de estados de alerta y posibles soluciones.

Para la adquisición de imágenes se utiliza el módulo de cámara que viene como accesorio y que está diseñado especialmente para Raspberry Pi, el cual se conecta mediante el puerto CSI (Camera Serial Interface), es compatible con el sistema operativo Raspbian y cuenta con una librería especializada para realizar aplicaciones en Python3. Este módulo tiene un tamaño de 2.5 x 2 x 1 cm y cuenta con un sensor de 5 Megapíxeles, con el que puede capturar imágenes con una resolución de hasta 2592 x 1944 píxeles, así mismo permite la grabación de video en diferentes formatos. En la Figura 40 se puede observar el dispositivo que se utilizó.

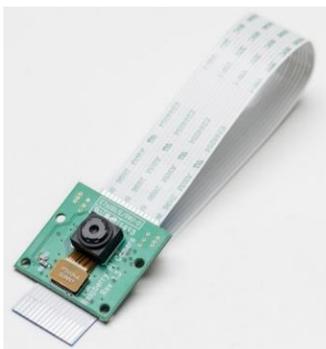


Figura 40. Módulo cámara Raspberry Pi

Fuente: [www.parallella.org](http://www.parallella.org)

---

<sup>3</sup> “Un meteoro es un fenómeno observado en la atmósfera o sobre la superficie de la tierra, que consiste en una suspensión, una precipitación, o un depósito de partículas líquidas, acuosas o no, o de partículas sólidas, o un fenómeno de la naturaleza de manifestación óptica o eléctrica.” [51] por ejemplo: neblina, tempestad de arena, arco iris, halo solar, relámpagos entre otros.

Para poder utilizar el módulo de cámara se utiliza la librería “picamera” desarrollada especialmente para aplicaciones con Python, lo cual permite la integración al código de la aplicación desarrollada para el prototipo, las imágenes se capturan seleccionando la ruta de destino y el nombre correspondiente, y con la ayuda de la librería se ajustan características de las imágenes como la velocidad de obturación, resolución, y brillo entre otras. Adicionalmente a la captura de fotografías se adhiere un segmento de código el cual permite colocar en las imágenes capturadas, una marca de tiempo con la fecha y hora de la captura con el fin de tener un registro de cada una de ellas. En el Código 1, se presenta el código utilizado para la captura de una imagen instantánea.

#### Código 1. Captura de imágenes “Foto.py”

```
#!/usr/bin/python
from picamera import PiCamera
from datetime import datetime
from subprocess import call

camera = PiCamera()

#Asigna el fecha y hora actual del sistema
actual = datetime.now()
formato1 = "%Y%m%d"
formato2 = "%H%M%S"
formato3 = "%Y/%m/%d"
formato4 = "%H:%M:%S"
fecha = actual.strftime(formato1)
hora = actual.strftime(formato2)
fecham = actual.strftime(formato3)
horam = actual.strftime(formato4)

try:
    #Captura de Imagenes
    fileName='/home/pi/Documents/ProyectoGrado/Fotos/'+fecha+'_'+hora+'.jpg'
    camera.resolution = (2592, 1944)
    camera.start_preview()
    camera.capture(fileName,resize=(520,380))
    camera.stop_preview()
    camera.close()

    #Marca de Tiempo
    timestampMessage = "Fecha: "+fecham+" Hora: "+horam
    timestampCommand = "/usr/bin/convert " + fileName + " -pointsize 16
    -fill yellow -annotate +275+370 '" + timestampMessage + "' " + fileName
    call([timestampCommand], shell=True)
except:
    camera.stop_preview()
    camera.close()
```

#### 4.7 Interfaz gráfica para configuración de parámetros y visualización de datos

Debido a que es importante configurar y visualizar los datos de los sensores medidos de forma local, se diseñó e implementó una aplicación apta para ejecutarse en el sistema operativo Raspbian, la cual permite configurar de forma amigable los parámetros de los sensores y mostrar el comportamiento de cada una de las variables.

Como interfaz de visualización, se usó una pantalla táctil ("*Raspberry Pi Touch Display*"), la cual cuenta con una longitud de 7", una resolución de 800x400 píxeles y un sistema *touchscreen* que la hace completamente interactiva. La conexión a la tarjeta Raspberry Pi se realiza a través de un cable de cinta flexible por medio del puerto DSI (*Display Serial Interface*), y se alimenta con 5V<sub>DC</sub>. Adicionalmente se adquirió la carcasa de protección para salvaguardar la tarjeta base y que permite la ubicación de la pantalla dentro del gabinete. En la Figura 41, se puede observar la pantalla y carcasa utilizadas para el prototipo.

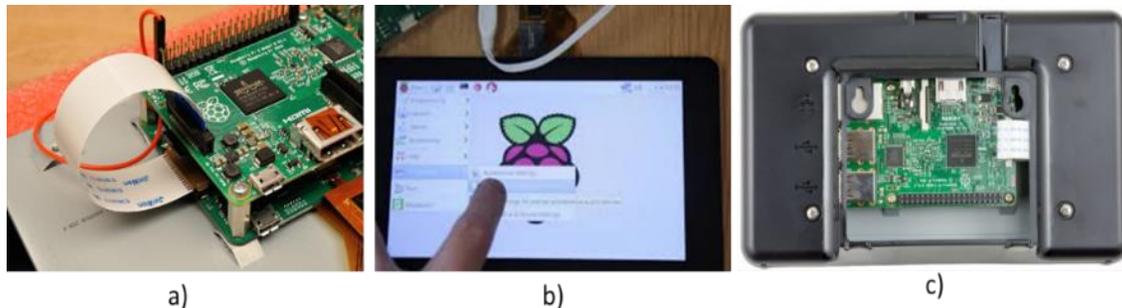


Figura 41. Pantalla y carcasa Raspberry Pi 7"  
a) Cinta DSI, b) Pantalla "touchscreen", c) Carcasa

Para el desarrollo de la aplicación gráfica se utilizó la plataforma de software **MONO**, la cual es la implementación de código fuente abierto del **.Net Framework** de Microsoft, que permite desarrollar y ejecutar aplicaciones multiplataforma en los sistemas operativos Linux, Windows, OSX, BSD y Solaris. **MONO** cuenta con un compilador de **C#**, el cual es un lenguaje de programación orientado a objetos utilizado para el desarrollo de la aplicación, cuenta también con una gran cantidad de librerías útiles para diferentes tipos de aplicaciones. Para la creación de la interfaz gráfica de visualización, se utiliza entre otras **GTK#**, que es la herramienta de mono y .net para entornos de aplicaciones GUI

(*Graphical User Interface*), la cual contiene los objetos y funciones para crear instrumentos como ventanas, botones, menús, etiquetas, pestañas, etc.

Para desarrollar la interfaz gráfica de configuración y visualización se utiliza el entorno de desarrollo integrado (IDE) **MonoDevelop**, el cual trabaja con mono y permite la compilación y ejecución de código en C# y otros lenguajes. Es un software gratuito de fácil instalación en sistemas operativos Linux, que posee características como el autocompletar código, el soporte para proyectos gráficos con la librería GTK#, y la capacidad de integrarse con aplicaciones como MySql por medio de complementos descargables. Adicionalmente, se cuenta con la ventaja de tener una gran variedad de documentación de acceso público para .NET y GTK#, lo cual es conveniente para el desarrollo de aplicaciones. En la Figura 42, se puede observar el entorno de desarrollo de MonoDevelop.

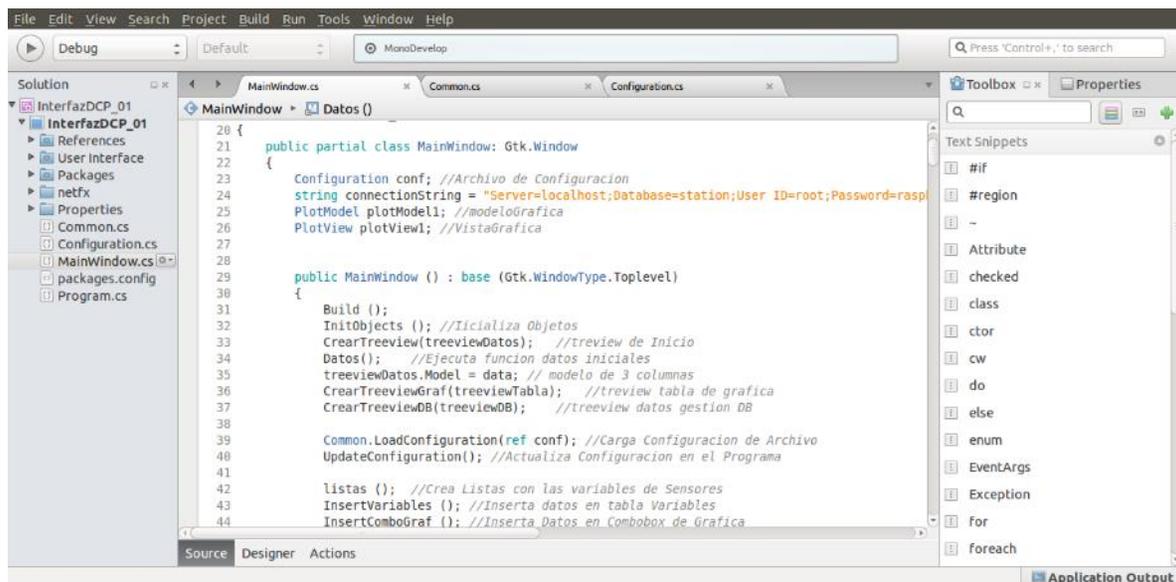


Figura 42. Captura de pantalla del sistema de desarrollo IDE MonoDevelop

La interfaz gráfica de visualización local está compuesta por pestañas o paneles con los que el usuario interactúa, permitiendo diferentes opciones para la configuración, visualización, almacenamiento y gestión de la información obtenida por los sensores hidrometeorológicos. A continuación se describe cada uno de los paneles que componen la interfaz gráfica.

### 4.7.1 Panel de configuración

En el panel de configuración, el usuario ingresa todos los datos necesarios para el funcionamiento adecuado de la adquisición y procesamiento de las variables, ingresando información asociada a la identificación, muestreo y parámetros de medición de acuerdo al tipo y características de los sensores a utilizar. Este panel cuenta con 4 secciones para el ingreso de diferente tipo de información: *configuración inicial*, *sensores análogos*, *sensores digitales*, *sensor SDI-12*.

En la **configuración inicial** el usuario ingresa los datos asociados a la identificación de la estación, **nombre y código**, también debe definir los valores para el muestreo y almacenamiento de información, **tiempo de visualización** y **tiempo de muestreo**, así mismo se debe precisar la activación o no de las funciones para **almacenar en BD** y **Activar Tx**, para la ejecución periódicamente estos procesos. En la Figura 43, se puede observar la sección de la interfaz de configuración que presenta los parámetros iniciales.



Figura 43. Panel de configuración - Configuración inicial

En las siguientes secciones del panel de configuración se debe tener en cuenta los parámetros de las variables hidrometeorológicas que se desean sensor, cuando se quiere seleccionar un canal para la medición se activa o desactiva el botón de verificación, donde

solo se tendrán en cuenta los canales con la casilla activa. Sin importar el tipo de sensor todas las variables deben tener **Nombre**, asociado al parámetro de medición, **ID**, número único de identificación, **Unidad**, dependiendo del sistema de medida, y número de **Decimales** con los que se quiere visualizar el dato.

En la sección de **sensores análogos** se debe asociar los parámetros correspondientes, como se muestra en la Figura 44, donde se deben colocar las casillas correspondientes a **InMin**, **InMax**, **OutMin**, **OutMax**, que como se explicó en la sección 4.4.1, se utilizan para la linealización de la variable con respecto al voltaje de entrada.

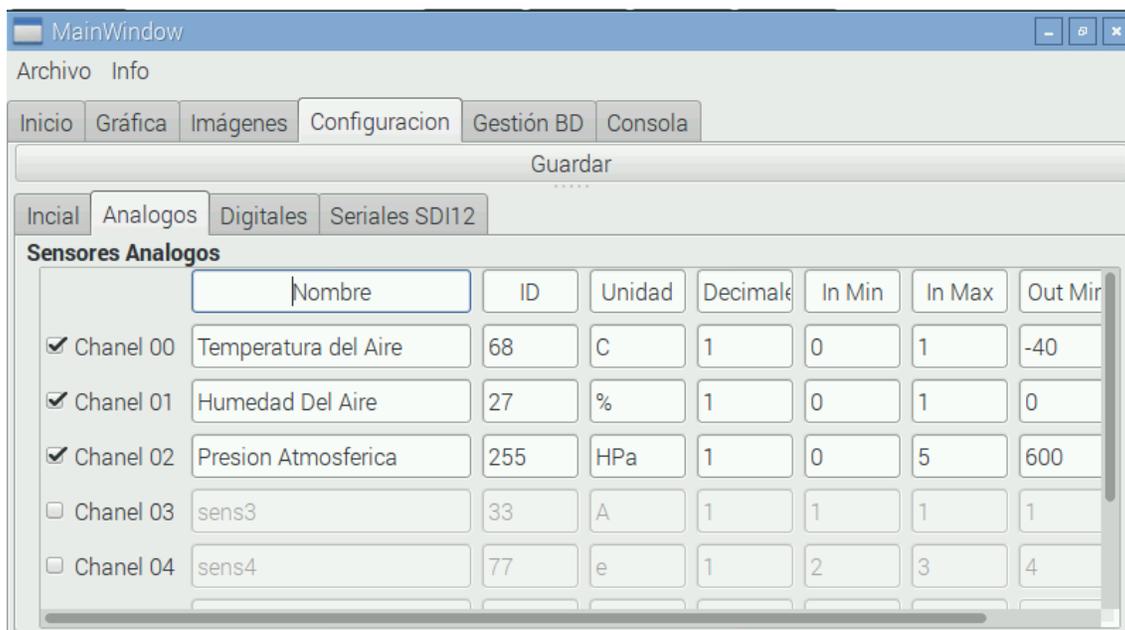


Figura 44. Panel de configuración - Entradas análogas

En la Figura 45, se observa la sección de **sensores digitales** del panel de configuración donde se puede escoger entre los dos canales disponibles, y se debe especificar la resolución asociada al pulso digital.

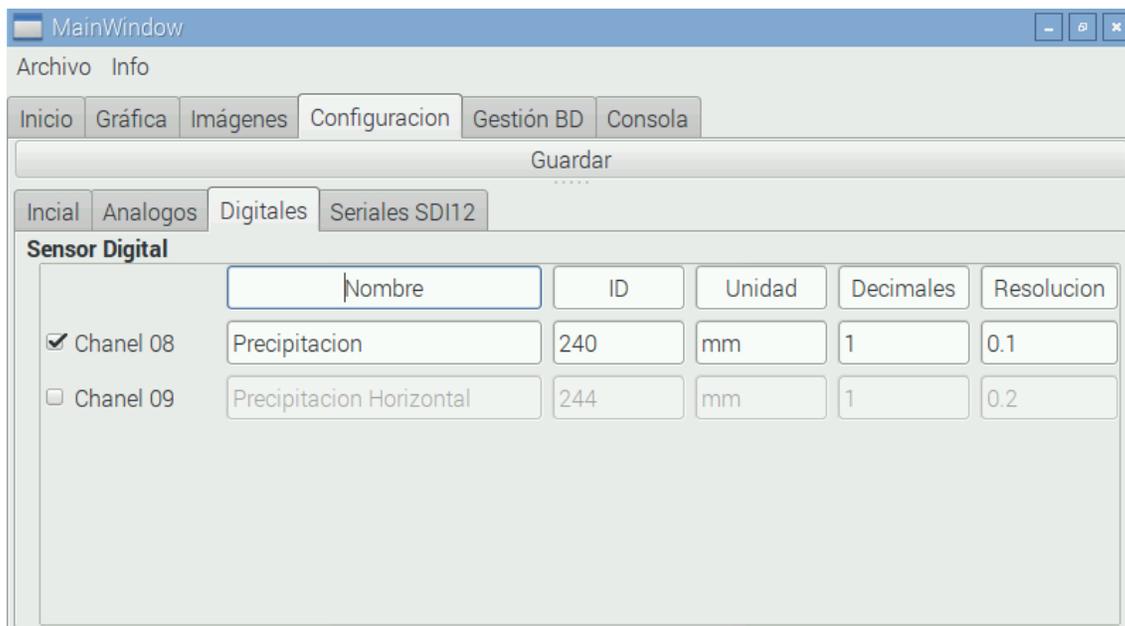


Figura 45. Panel de configuración - Entradas digitales

La Figura 46, muestra la sección de configuración de la entrada SDI-12 en la cual se puede seleccionar hasta 2 variables diferentes que se miden por el mismo canal, especificando el número de esclavo correspondiente al sensor y la posición del dato a seleccionar dentro del buffer de respuesta del sensor.



Figura 46. Panel de configuración - Entrada serial SDI-12

#### 4.7.1.1 Archivo de configuración XML

Toda la información registrada por el usuario en el panel de configuración se guarda en un archivo de texto plano en formato XML (*eXtensible Markup Language*), el cual es un formato estándar abierto que es muy popular y cualquier aplicación puede realizar su procesamiento tanto de lectura como escritura. En el Código 2, se observa un segmento del archivo de configuración generado por la interfaz gráfica y que posteriormente es procesado por la aplicación base en Python3 para realizar el proceso de medición de variables hidrometeorológicas cómo se explicó en la sección 4.4.

#### Código 2. Ejemplo del archivo de configuración XML

```
<?xml version="1.0"?>
<Configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PosVpanedA>103</PosVpanedA>
  <PosVpanedB>222</PosVpanedB>
  <PosVpanedC>143</PosVpanedC>
  <PosVpanedDB>462</PosVpanedDB>
  <NombreEstacion>ECI Julio Garavito</NombreEstacion>
  <CodigoEstacion>88112901</CodigoEstacion>
  <EstadoCH0>true</EstadoCH0>
  <NombreCH0>Temperatura del Aire</NombreCH0>
  <IDCH0>68</IDCH0>
  <UnidadCH0>C</UnidadCH0>
  <DecimalesCH0>1</DecimalesCH0>
  <InMinCH0>0</InMinCH0>
  <InMaxCH0>1</InMaxCH0>
  <OutMinCH0>-40</OutMinCH0>
  <OutMaxCH0>60</OutMaxCH0>
```

#### 4.7.2 Panel de inicio

El panel de inicio es el espacio para visualizar en tiempo real las variables adquiridas por los sensores activos (Figura 47 (a)), estas variables se actualizan de acuerdo a los nombres, unidad e id seleccionados en el panel de configuración.

Para empezar a visualizar el dato adquirido se debe activar la casilla **Iniciar Captura** y su valor se actualiza de acuerdo al tiempo de visualización establecido en el panel de configuración, como se ve en la Figura 47 (b).

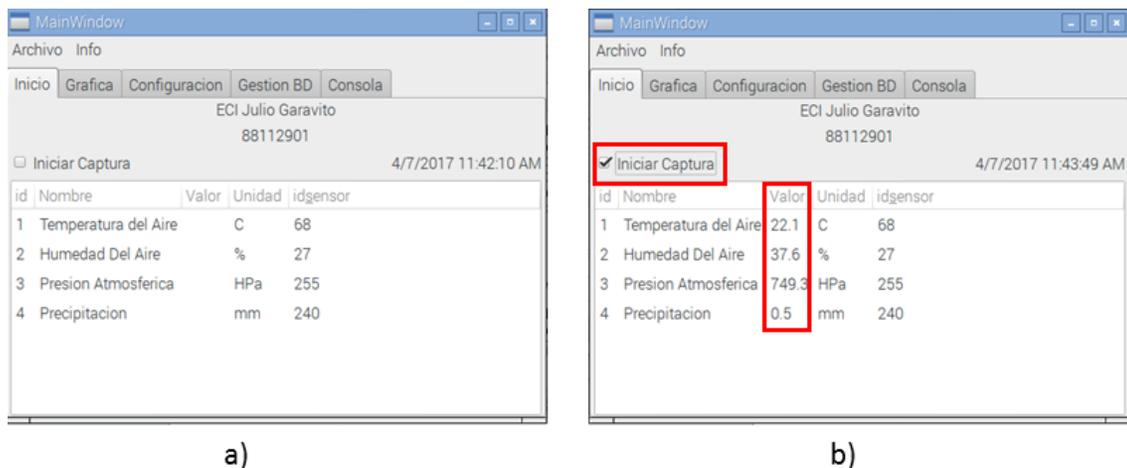


Figura 47. Panel de inicio  
a) Estado Inicial, b) Captura y visualización de Variables en tiempo real.

### 4.7.3 Panel de visualización gráfica de variables

EL panel de generación gráfica está diseñado para realizar la visualización y análisis del comportamiento de las variables previamente adquiridas con respecto al tiempo, los datos son obtenidos del almacenamiento en la base de datos y sirven para revisar el comportamiento de una variable o para diagnosticar el funcionamiento de los sensores hidrometeorológicos. La presentación de la información se realiza de forma gráfica y de forma tabular, permitiendo seleccionar cualquiera de las diferentes variables que están configuradas dentro de los parámetros de la estación. En la Figura 48, se puede observar el panel de generación gráfica.

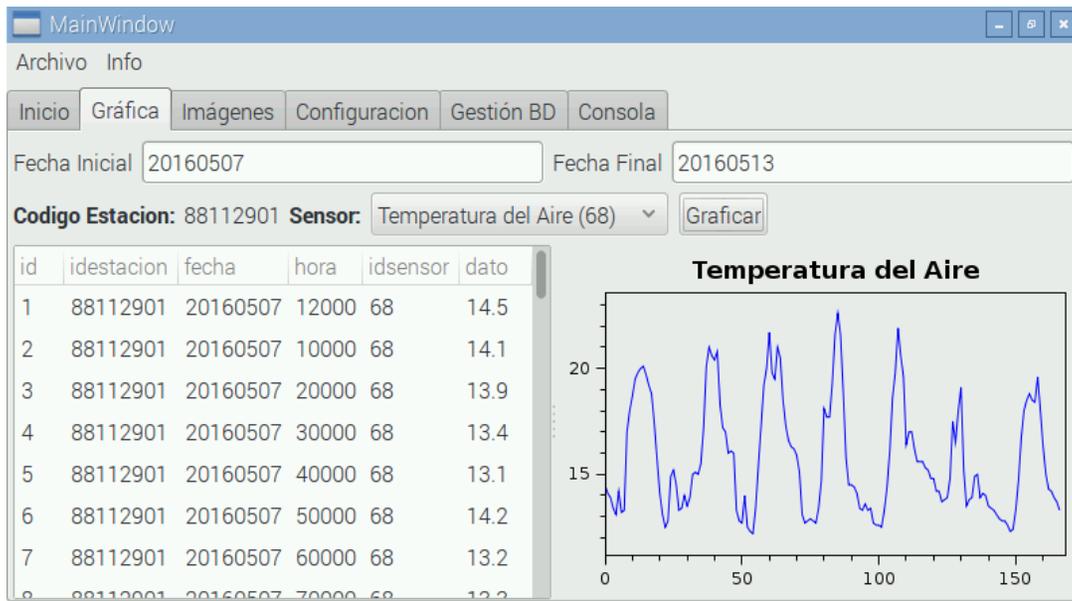


Figura 48. Panel de visualización gráfica de variables

#### 4.7.4 Panel gestión base de datos

En el panel de gestión de la base de datos, se pueden realizar consultas asociadas a los parámetros almacenados en la base de datos por medio de comandos básicos SQL. En la Figura 49 se puede observar este panel de gestión.

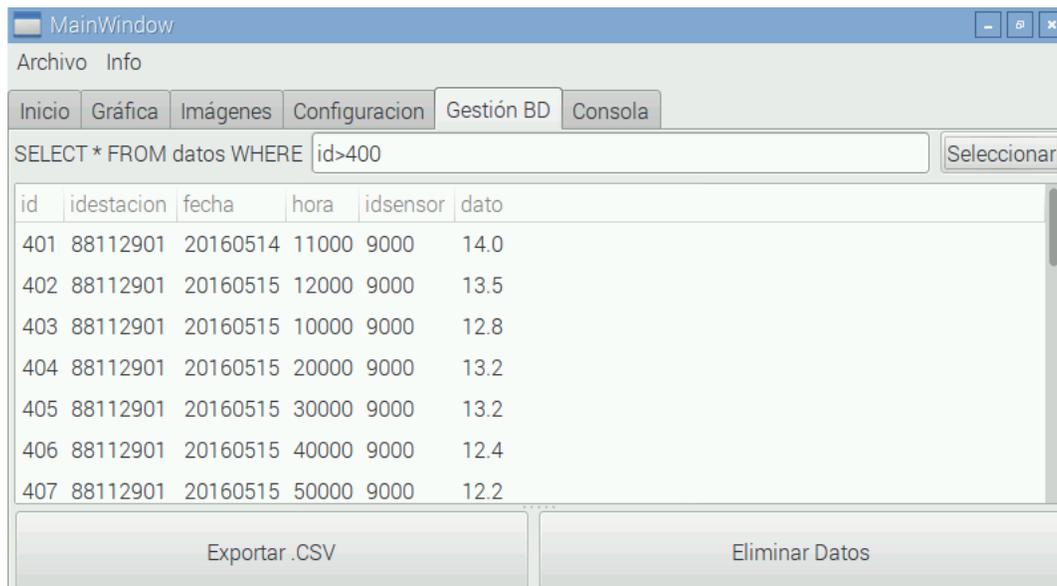


Figura 49. Panel de gestión base de datos

Dentro del panel de gestión existe la posibilidad de realizar la exportación de la información consultada a un archivo de texto plano con un formato de columnas separado por el caracter punto y coma “;”, el cual tendrá una extensión “.CSV” y quedará almacenado en el escritorio del sistema operativo. Este tipo de archivos se puede abrir fácilmente con programas ofimáticos de manejos de hojas de cálculo como Excel o LibreOfficeCalc. En la Figura 50 se puede observar el archivo exportado en el escritorio y la forma de decodificación en el programa LibreOfficeCalc.

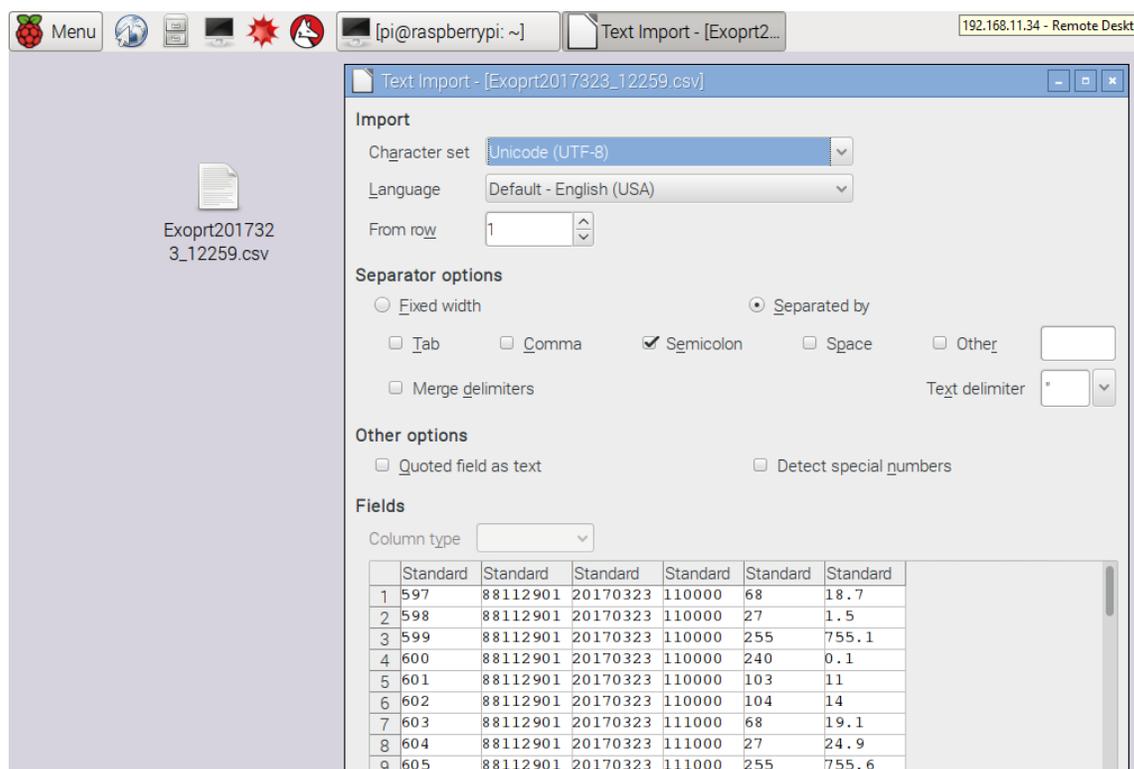


Figura 50. Archivo “.CSV” exportado de la interfaz

Otra funcionalidad con la que se cuenta en el panel de gestión de la base de datos es la posibilidad de eliminar registros almacenados en la base de datos, teniendo en cuenta que como elemento de prevención ante la posibilidad de presionar equivocadamente el botón, el software preguntará y confirmará el borrado de los datos como se muestra en la Figura 51, posterior a la confirmación afirmativa se eliminan los registros seleccionados de la base de datos.

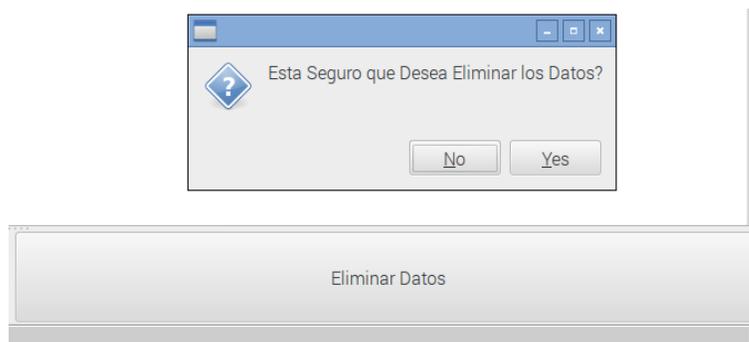


Figura 51. Ventana de confirmación del proceso que permite eliminar registros base de datos

#### 4.7.5 Panel de visualización de imágenes

El panel de visualización de imágenes se divide en dos secciones, la primera sección se utiliza para el proceso de captura de imágenes, el usuario puede realizar una captura instantánea presionando un botón, o puede realizar la programación de capturas periódicas con la posibilidad de tomar hasta 3 imágenes diarias especificando la hora y el minuto en el que desea realizar la captura. La segunda sección se utiliza para visualizar las imágenes adquiridas, cuando el usuario presiona el botón “**seleccionar**”, se despliega un explorador de archivos, en el cual debe seleccionar el origen de la imagen que desea visualizar, la cual es mostrada en la pantalla de la interfaz. En la Figura 52, se observa el panel de visualización de imágenes.

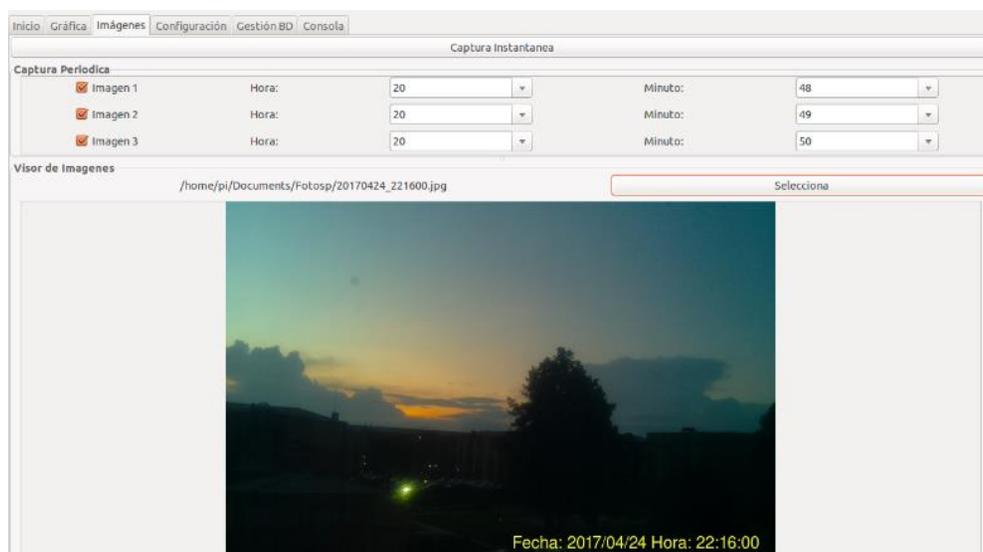


Figura 52. Panel de visualización de imágenes

#### 4.8 Sistema de comunicación inalámbrica

Dada la necesidad de poder centralizar la información de los datos registrados por los diferentes sensores hidrometeorológicos, se decidió utilizar un sistema de transmisión inalámbrica que envíe dicha información a un servidor remoto del IDEAM. Debido a que en Colombia la cobertura de señal celular está presente en la mayoría de centros poblados, se utiliza el sistema de comunicación GPRS (*General Packet Radio Service*). Para esta etapa del proyecto se utilizó el módem celular **MOXA OneCell G3151** como el mostrado en la Figura 53, el cual es un módem de tipo industrial que funciona con un voltaje de 9 a 48V y que opera en las bandas de 900/1800, 850/1900 MHz, utilizando una tarjeta SIM con un plan de datos celular convencional. El módem permite el registro a la red GPRS, acceso a internet y el envío de información a través de diferentes protocolos, este módem incorpora un puerto Ethernet con el cual se puede realizar su configuración y con el que además se conecta a la Raspberry Pi.



Figura 53. Módem celular Moxa OnCellG3151

Fuente: [www.moxa.com](http://www.moxa.com)

Este módem cuenta con un servidor web interno con el cual se realiza la configuración de los diferentes parámetros como el rango de IP, el tipo de conexión, el sistema de nombre de dominio (DNS), el nombre del punto de acceso (APN) del operador celular y se pueden visualizar variables de estado del módem como los niveles de señal. En la Figura 54, se muestra la vista principal de la interfaz de configuración del módem moxa OnCellG3151.

The screenshot shows the Moxa OnCell G3151 web interface. On the left is a navigation menu with categories like Main Menu, Overview, Basic Settings, Network Settings, and System Management. The main content area displays a 'Welcome to OnCell G3151' message followed by a table of system parameters.

Welcome to OnCell G3151	
Model name	OnCell G3151
Serial No.	7692
Firmware version	1.4 Build 13020611
WAN IP address	10.17.184.195
Ethernet IP address	192.168.127.254
Ethernet MAC address	00:90:E8:3D:84:A8
Ethernet LAN speed	100M/Link
Cellular signal level	3
Cellular signal RSSI	24
Cellular mode	GPRS
Cellular IMEI	354037057393542
Up time	0 days 00h:01m:44s
Serial port 1	115200,None,8,1,RS-232

Figura 54. Configuración módem celular Moxa OnCellG3151

Así mismo el módem cuenta con indicadores luminosos tipo *led* para la verificación externa del estado de la conexión, en la Figura 55, se puede observar los *leds* del módem donde se indica el nivel de señal alto y la conexión por GPRS activa.



Figura 55. Indicadores de estado módem celular Moxa OnCellG3151

#### 4.9 Integración de datos al sistema de visualización meteorológico Hydras3

Con el fin de visualizar de forma remota los datos capturados por los sensores, se hace la integración con el sistema nacional de visualización de estaciones meteorológicas

automáticas del Instituto de Hidrología Meteorología y Estudios Ambientales (IDEAM), conocido como **Hydras3**, descrito en la sección 3.7.

Para realizar la integración de los datos, es necesario la creación de un archivo de texto que contenga la información de los sensores en formato .MIS así mismo se necesita una plantilla en formato XML con la configuración de las variables de la estación. A continuación se describen los formatos de estos dos archivos.

#### 4.9.1 Archivo de datos “.MIS”

El sistema de visualización Hydras3 realiza la lectura de los valores generados por las diferentes estaciones en un archivo de texto plano con extensión “.MIS”, este archivo se debe colocar en una ubicación específica dentro del servidor del IDEAM utilizando el protocolo **FTP** (*File Transfer Protocol*) y debe cumplir con un formato y una identificación correspondiente. En la Figura 56 se muestra el formato de presentación del archivo .MIS, en esta imagen se observa cómo debe estar compuesto el nombre del archivo, especificando el código de la estación, la fecha y la cantidad de datos a enviar.



Figura 56. Presentación archivo “.MIS”

Fuente: IDEAM

La Figura 57 muestra la estructura interna del archivo “.MIS”, la cual está compuesta por el **código de la estación**, que es un código único suministrado por el IDEAM de acuerdo al catálogo nacional de estaciones, un **código interno de sensor** el cual es seleccionado de acuerdo a las variables establecidas por el IDEAM, la **fecha y hora** de ocurrencia en la

que se realizó la medición de la variable, y el **valor del dato** correspondiente a la información adquirida por el sensor. Todos estos componentes son elementos de texto que deben ser ubicados en su orden específico.

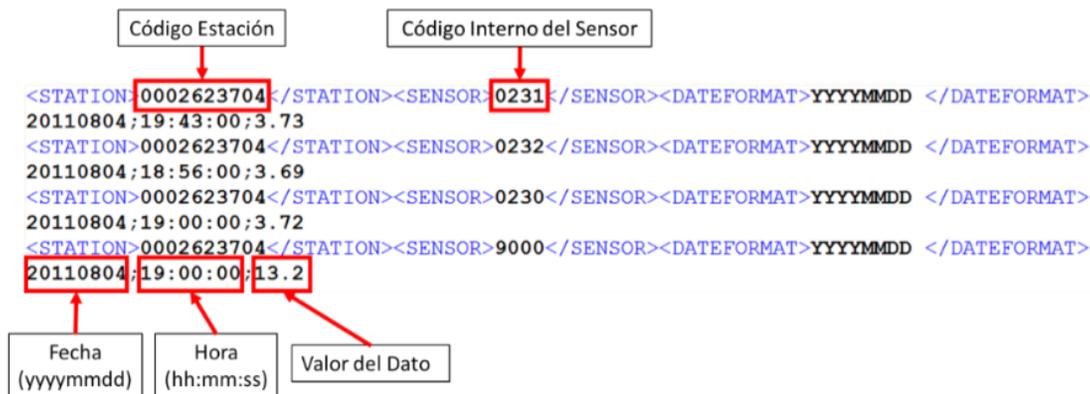


Figura 57. Estructura interna archivo “.MIS”

Fuente: IDEAM

Para la creación del archivo .MIS, en el prototipo se implementa una función en Python3, la cual toma los datos de cada una de las variables y la información entregada por el usuario, para crear el archivo de texto plano en el formato explicado anteriormente, para luego ser enviado vía internet al servidor FTP, de acuerdo al tiempo de muestreo establecido por el usuario. En el Código 3, se muestra el código de la función para la creación del archivo .MIS, en el ANEXO F, se encuentra en detalle la implementación y uso del código.

```

Código 3. Función crear archivo “.MIS”
#Funcion Crear .MIS
def crearMIS(id_estacion, id_sensor, datos, fecha, hora):
    id_estacion_str='00'+str(id_estacion)
    id_sensor_str=[]
    datos_str=[]
    for i in range(len(datos)):
        datos_str.append(str(datos[i]))
    Archivo = id_estacion_str+'_'+fecha+ hora+'.MIS' #Nombre del Archivo
    fo = open('/home/pi/Documents/ProyectoGrado/FTP/'+ Archivo , "w+") #Ruta
    para Crear El Archivo
    print ("Creando Archivo..." )
    for i in range(len(datos)):
        fo.write('<STATION>'+ id_estacion_str+'</STATION><SENSOR>'+
id_sensor_str[i] +'</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>\n')
        fo.write(fecha+ ';' + hora+ ';' + datos_str[i]+ '\n')
    fo.close() # Finaliza Creacion del Archivo
    print ("Archivo Local Creado" )
    return Archivo

```

## 4.9.2 Plantilla de la estación XML

El software Hydras3 requiere para la correcta visualización de los datos, una plantilla predefinida, la cual debe contener la estructura de la información a recibir, basado en esta plantilla el software identifica los valores correspondientes para almacenar y mostrar la información de forma ordenada. Este archivo es un documento de texto plano en formato XML, el cual tiene una serie de etiquetas y contenidos que están asociados a datos como el código de la estación, nombre de la estación, código del sensor, nombre y unidad de la variable, entre otros. Dependiendo del número de variables que tenga la estación se crean cuantas etiquetas sea necesario. En la Figura 58, se observa un ejemplo de la estructura interna de la plantilla utilizada para la visualización de los datos del prototipo en Hydras3. En el ANEXO F , se muestra la plantilla completa implementada.

```
<STATION>
  <DEVICETYPE>RASPBERRY PI 2</DEVICETYPE>
  <STATIONID>0088112901</STATIONID>
  <STATIONNAME>ECI JULIO GARAVITO</STATIONNAME>
  <STATIONUTC>-05:00</STATIONUTC>
</SENSOR>
  <SENSORID>0068</SENSORID>
  <SENSORNAME>Temperatura del Aire</SENSORNAME>
  <UNIT>°C</UNIT>
  <RIGHTDIGITS>1</RIGHTDIGITS>
  <AXISLABEL>Temperatura del Aire</AXISLABEL>
  <INTERVAL>0</INTERVAL>
</SENSOR>
</STATION>
```

The diagram shows the XML structure for a station and a sensor. Red arrows point from the XML tags to their corresponding descriptions:

- `<DEVICETYPE>RASPBERRY PI 2</DEVICETYPE>` → Tipo de Dispositivo
- `<STATIONID>0088112901</STATIONID>` → Código Estación
- `<STATIONNAME>ECI JULIO GARAVITO</STATIONNAME>` → Nombre Estación
- `<STATIONUTC>-05:00</STATIONUTC>` → Hora UTC
- `<SENSORID>0068</SENSORID>` → Código de Sensor
- `<SENSORNAME>Temperatura del Aire</SENSORNAME>` → Nombre de la Variable
- `<UNIT>°C</UNIT>` → Unidad de la Variable
- `<RIGHTDIGITS>1</RIGHTDIGITS>` → Dígitos Decimales
- `<AXISLABEL>Temperatura del Aire</AXISLABEL>` → Nombre de la Variable
- `<INTERVAL>0</INTERVAL>` → Intervalo de Recepcion

Figura 58. Estructura plantilla XML para Hydras3

Fuente: IDEAM

## 4.10 Sistema de alimentación

Para la alimentación del prototipo desarrollado y los sensores hidrometeorológicos empleados, se utiliza la unidad de control de potencia **OTT-PCU12**, la cual es una fuente de alimentación regulada de corriente continua con salida a 12V, diseñada especialmente para aplicaciones de hidrología y meteorología. La **PCU12** puede funcionar tanto conectada a la red eléctrica convencional (90–250V<sub>AC</sub>) como con paneles solares, convirtiéndose en un sistema de alimentación solar fotovoltaico, ya que funciona como un

regulador solar de carga a 12 Amperios. En la Figura 59, se puede observar los tipos de conexión de la PCU12.

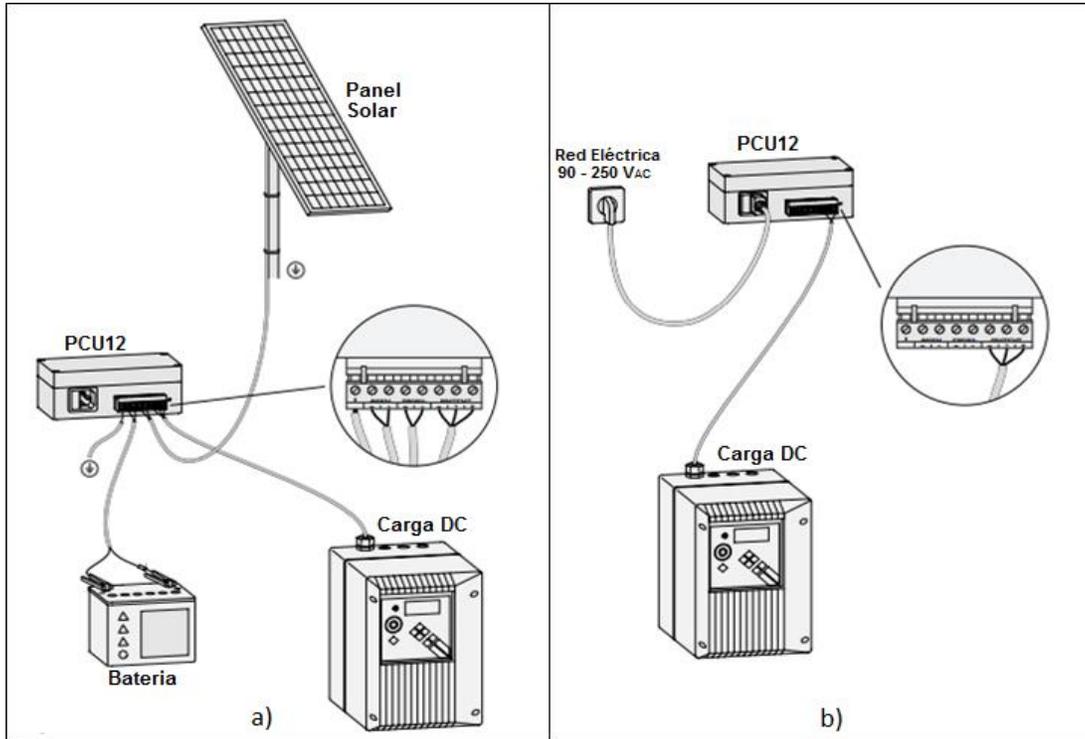


Figura 59. Tipos de conexión unidad de control de potencia - PCU12

a) Sistema de alimentación solar, b) Red eléctrica

Fuente: Hoja Técnica OTT - PCU12

En el ANEXO D, se realiza el estudio del dimensionamiento de un sistema solar fotovoltaico para el prototipo desarrollado, donde en base al consumo de potencia de cada uno de los componentes, se determinó el uso de un **panel solar de 80W** y una batería de **12V a 110Ah**, los cuales conectados al **regulador PCU12**, permite el funcionamiento adecuado del prototipo.

## 5 VERIFICACIÓN DE FUNCIONAMIENTO

### 5.1 Funcionamiento entradas análogas

Con el fin de verificar el funcionamiento y calidad de la lectura de señales de voltaje, se realizó el cálculo del error y la incertidumbre en las entradas análogas del prototipo desarrollado, tomando como base la directriz de la **Asociación Europea de Institutos Nacionales de Metrología (EURAMET)** sobre calibración de multímetros digitales [52], así como el procedimiento para la calibración de multímetros digitales [53] del **Centro Español de Metrología (CEM)**. Esto debido a que el prototipo desarrollado, al igual que las plataformas colectoras de datos y los multímetros digitales, cuentan con un conversor análogo digital y circuitos electrónicos auxiliares como sistema básico de medida.

El procedimiento de verificación consiste en realizar la comparación de valores generados por un instrumento patrón certificado y la lectura de la medición hecha por el prototipo desarrollado, por lo cual en la guía de la EURAMET se hace referencia a la selección de ciertos puntos de medición en los que se debe tomar las muestras para realizar dicha comparación. En la Tabla 13, se muestran los porcentajes de selección para los puntos de medición de acuerdo a las diferentes magnitudes eléctricas.

Function	Measurement points <sup>(a)</sup>	
DC voltage	Zero <sup>(b)</sup> /+10 %, ±90 %	in all ranges
	±10 %, +50 %, ±90 % <sup>(c)</sup>	in a selected intermediate range
DC current	Zero <sup>(b)</sup> , +90 %	in all ranges
	+10 %, 50 % <sup>(d)</sup> , ±90 %	in a selected intermediate range
DC resistance	Zero <sup>(b)</sup> /+10 %, +90 %	in all ranges
AC voltage	10 %	50 Hz/1 kHz <sup>(e)</sup>
	10 %, 50 %, 90 %	20 Hz <sup>(f)</sup> , 50Hz, 1 kHz
AC+DC voltage	90 %	20 Hz <sup>(f)</sup> , 50 Hz, 1 kHz, 20 kHz, 50 kHz, 100 kHz <sup>(g)</sup>
AC current	10 %	50 Hz/1 kHz <sup>(e)</sup>
	90 %	20 Hz, 50 Hz, 1 kHz, 5 kHz <sup>(g)</sup>
Others <sup>(i)</sup>	Zero <sup>(b)</sup> /10 %, 90 %	in all ranges

Tabla 13. Definición puntos de medición para calibración de multímetros digitales

Fuente: [36]

En la tabla anterior se puede observar que para mediciones de voltaje en corriente directa, se debe tomar un total de 5 puntos correspondientes a los valores de  $\pm 10\%$ ,  $+50\%$  y  $\pm 90\%$  en el rango intermedio seleccionado. En vista que el prototipo desarrollado está diseñado únicamente para realizar mediciones en escala positiva, y a fin de mantener los 5 puntos de medición recomendados por las normas internacionales, se decide utilizar los valores mostrados en la Tabla 14, como puntos de medición para la verificación de funcionamiento en el rango de 0 a 5 voltios.

Porcentaje	Punto de Medición
10%	0,500 V
30%	1,500 V
50%	2,500 V
70%	3,500 V
90%	4,500 V

Tabla 14. Puntos de medición para verificación de funcionamiento.

Como instrumento patrón se utilizó el **Calibrador de Procesos Multifunción Fluke 725**, [54], de propiedad del IDEAM, con número de serie 3529050 y certificado de calibración expedido el 03 de junio de 2017, el cual es una herramienta portátil capaz de medir y generar señales calibradas en variables como tensión (V), corriente (mA), y resistencia ( $\Omega$ ), entre otros, con el cual se puede diagnosticar, calibrar y verificar instrumentos y variables de tipo industrial. En la Tabla 15, se puede observar las características técnicas de las funciones de generación (fuente) y medición del calibrador Fluke 725 en diferentes magnitudes eléctricas.

Función	Medición	Fuente
V cc	0 V a 30 V	0 V a 10 V
mA cc	0 a 24 mA	0 a 24 mA
Frecuencia	1 CPM a 10 kHz	1 CPM a 10 kHz
Resistencia.	0 $\Omega$ a 3200 $\Omega$	15 $\Omega$ a 3200 $\Omega$
Termopar	Tipos E, J, K, T, B, R, S, L, U, N, mV	
RTD (detectores termométricos de resistencia)	Pt100 $\Omega$ (385) Pt100 $\Omega$ (3926) Pt100 $\Omega$ (3916) Pt200 $\Omega$ (385) Pt500 $\Omega$ (385) Pt1000 $\Omega$ (385) Ni120	

Tabla 15. Resumen de las funciones de fuente y medición – Fluke 725

Fuente:[54]

El calibrador de procesos multifunción Fluke 725, se emplea como generador de voltaje en corriente continua, con el cual se obtiene el valor de medición real de cada uno de los puntos de medición seleccionados, conectando de forma directa la salida del calibrador a la entrada análoga del prototipo. En la Figura 60, se puede observar el diagrama de conexión del calibrador como fuente de voltaje conectado a un dispositivo de lectura.

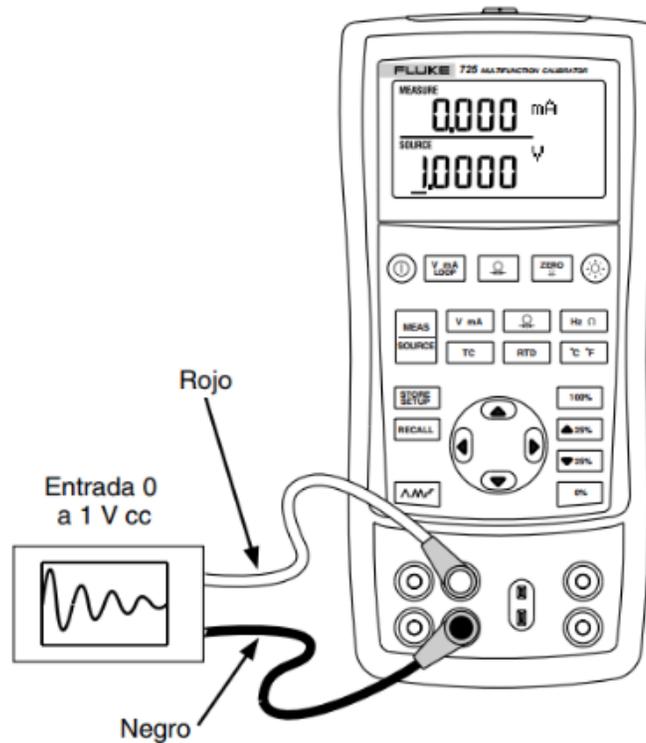


Figura 60. Conexión del calibrador como fuente de voltaje.

Fuente:[54]

Para cada uno de los puntos de medición seleccionados ( $\bar{X}$ ) se realiza una toma de 3 muestras ( $X$ ) con las cuales se realiza el cálculo del error absoluto, para posteriormente hacer la estimación de la incertidumbre básica para cada uno de los puntos de medición seleccionados. En la Tabla 16 se observan los resultados obtenidos del proceso de medición en el rango de operación del prototipo DCP desarrollado. De esta tabla se puede concluir que para los rangos de voltaje inferior la incertidumbre tiende a ser mínima, sin embargo a medida que el voltaje aumenta acercándose al valor mayor del rango, la incertidumbre toma un valor considerable que afecta la medida significativamente.

Punto de Medicion	Valor Medido	Error Absoluto	Incertidumbre
$\bar{X}$	$X$	$E =  X - \bar{X} $	$S = \sqrt{\frac{\sum(X - \bar{X})^2}{n - 1}}$
0,500	0,519	0,019	±0,030
0,500	0,522	0,022	
0,500	0,531	0,031	
1,500	1,564	0,064	±0,074
1,500	1,561	0,061	
1,500	1,555	0,055	
2,500	2,615	0,115	±0,141
2,500	2,61	0,116	
2,500	2,615	0,115	
3,500	3,636	0,136	±0,151
3,500	3,611	0,111	
3,500	3,622	0,122	
4,500	4,692	0,192	±0,236
4,500	4,694	0,194	
4,500	4,693	0,193	

Tabla 16. Resultados de mediciones y cálculo de Incertidumbre

Otro concepto que se tiene en cuenta para la verificación del funcionamiento y comportamiento de las entradas análogas, es la correlación existente entre los valores generados por el calibrador y los valores capturados por las entradas análogas del prototipo. Con los mismos datos obtenidos en la Tabla 16, y utilizando **Excel**, se realiza el análisis de regresión, obteniendo el gráfico mostrado en la Figura 61, en la cual se puede observar que se tiene una relación lineal positiva determinada por la función  $y = 1,0401x + 0,0019$ . Adicionalmente, se calcula el coeficiente de determinación ( $R^2=0,9999$ ), indicando un 99% de relación entre los voltajes generado y los voltajes capturados.

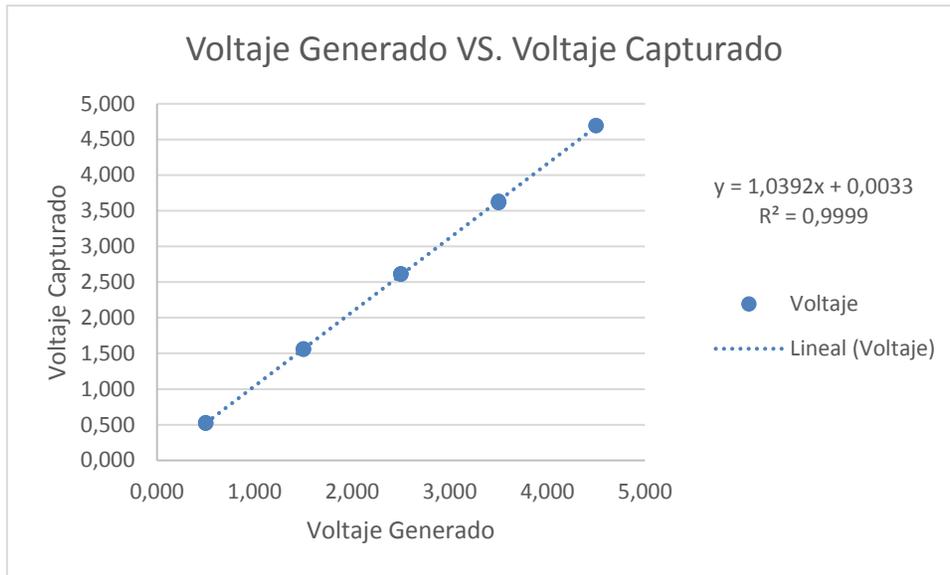


Figura 61. Correlación de datos medidos

## 5.2 Funcionamiento entradas digitales

Para la verificación del funcionamiento de las entradas digitales por pulsos se hace uso de un instrumento manual de propiedad del IDEAM como el mostrado en la Figura 62, el cual está compuesto por un interruptor y un pulsador, que al ser presionados crean un corto en los terminales de salida generando un pulso digital, simulando el comportamiento de los medidores de lluvia de tipo balancín (tipping bucket rain gauge) como el sensor de precipitación **RIMCO 8500** que fue descrito en la sección 3.6.3.



Figura 62. Generador de pulsos digitales

Fuente: IDEAM

El generador de pulsos se conecta directamente a cada una de las entradas digitales del prototipo y aleatoriamente se producen cierta cantidad de pulsos de forma manual, al realizar la comparación entre pulsos generados manualmente contra los pulsos que son leídos por el prototipo, se evidencia que la correspondencia de los datos es del 100%, es decir los pulsos generados manualmente son exactamente igual a los pulsos registrados en la interfaz gráfica. En la Figura 63, se muestra un ejemplo del experimento realizado, donde en una entrada digital se capturan 6 pulsos y en la segunda entrada se capturan 4 pulsos utilizando el generador manual.



Figura 63. Verificación de pulsos leídos por el prototipo.

### 5.3 Funcionamiento entradas serial SDI-12

Para verificar el funcionamiento de la entrada serial SDI-12, es importante revisar la ejecución del protocolo SDI-12 entre el prototipo desarrollado y el sensor de dirección y velocidad del viento **Vaisala WMT702**, descrito en la sección 3.6.4, por lo cual se decide utilizar un analizador de protocolos (sniffer), el cual se implementa utilizando la placa de desarrollo **Arduino UNO** y la librería **Arduino-SDI-12**. La conexión física del analizador de protocolos se muestra en la Figura 64, donde el Arduino UNO intercepta la línea datos (Serial Data Line) y la referencia de tierra (GND Line) con el fin de capturar la trama de datos y visualizarla en un computador.

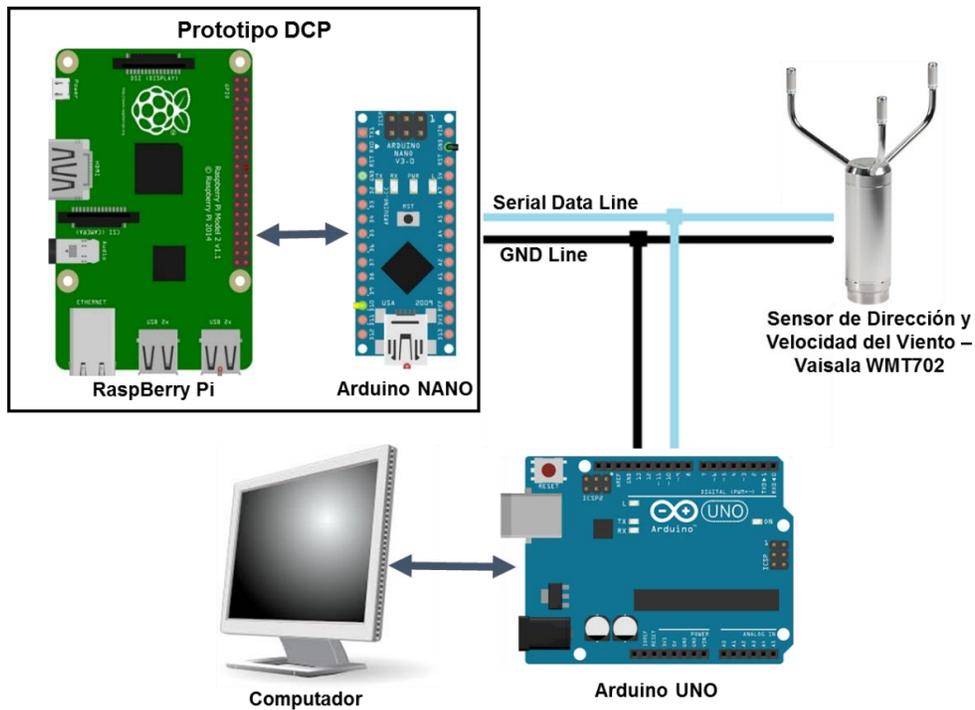


Figura 64. Conexión del analizador de protocolo SDI-12 utilizando Arduino UNO

Desde un computador por medio de cualquier terminal serial, es posible observar la trama de datos capturados por el analizador de protocolo, en la Figura 65, se muestra la captura de pantalla correspondiente a la conversación establecida entre el prototipo y el sensor mediante el protocolo SDI-12, en la cual se observa los comandos correspondientes al llamado y respuesta del sensor tal como se explicó en la sección 4.4.3.

En la Figura 66 a), se muestra la conversación del protocolo SDI-12 para el sensor de vientos utilizado, la cual empieza con el comando de inicio de medición (**0M!**), seguido de la respuesta del sensor (**00025**), indicando 2 segundos de demora y 5 datos de respuesta, a continuación se envía el comando de lectura de datos (**0D0!**) al cual el sensor responde con la trama de datos característica del sensor (**0+1.7+340.5-2.2+1.1+772.8**) donde de acuerdo a lo explicado en la sección 3.6.4, la posición 1 corresponde a la velocidad del viento (**1.7**) y la posición 2 a su dirección (**340.5**). Una vez comprobada la comunicación exitosa por medio del protocolo SDI-12 se verifica la lectura de los datos en la interfaz gráfica desarrollada la cual es mostrada en la Figura 66 b). La secuencia de lectura se repite continuamente dependiendo del tiempo de lectura programado por el usuario.

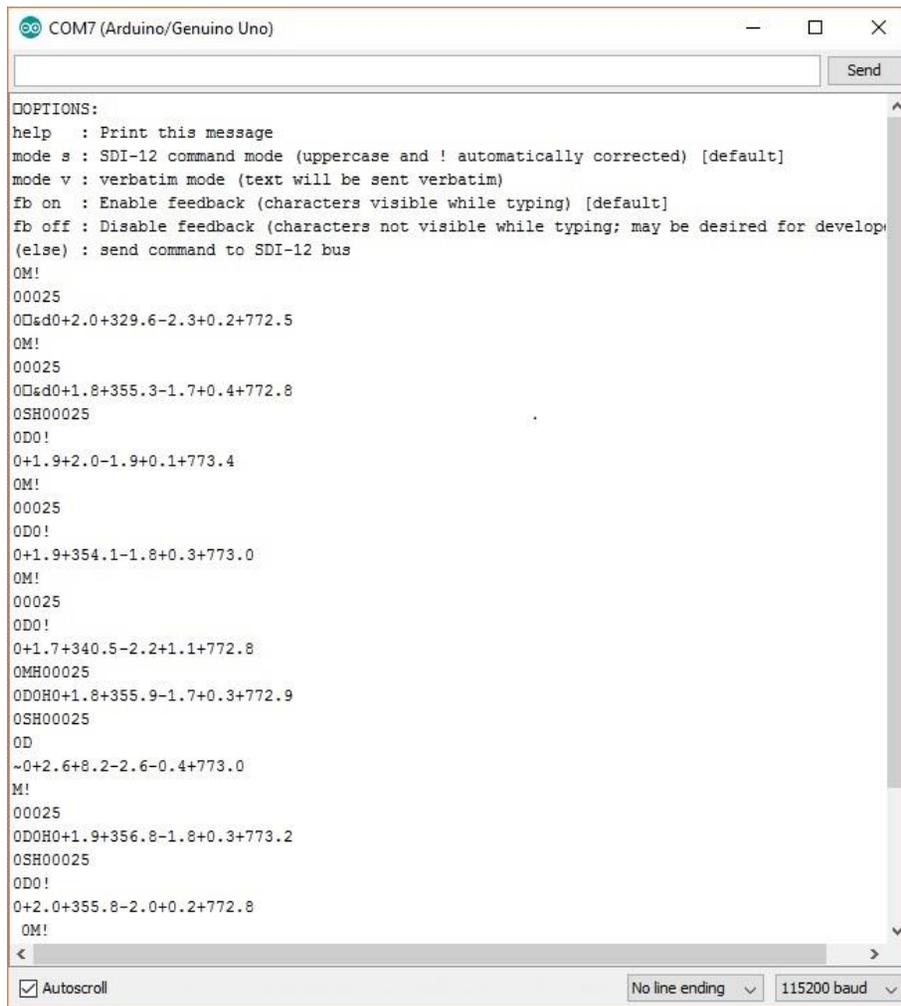


Figura 65. Captura de pantalla de la trama de datos capturada por el analizador de protocolo SDI-12

```

OM!
00025
OD0!
O+1.7+340.5-2.2+1.1+772.8
OMH00025
OD0H0+1.8+355.9-1.7+0.3+772.9
OSH00025
OD
~0+2.6+8.2-2.6-0.4+773.0

```

a)

id	Nombre	Valor	Unidad	idsensor
1	Velocidad del Viento	1.7	m/s	103
2	Direccion del Viento	340.5	°N	104

b)

Figura 66. Verificación de lectura de datos SDI-12.

a) Respuesta del analizador de protocolo. b) Captura de datos en la interfaz gráfica.

## 6 RESULTADOS Y CONTRIBUCIÓN

### 6.1 Fabricación de circuitos impresos

Para el diseño de los circuitos impresos se utilizó el software **Proteus Design Suite**, el cual cuenta con gran variedad de encapsulados de componentes electrónicos para la fabricación de placas impresas, este software realiza las conexiones entre los componentes utilizados de forma automática, lo cual facilita la fabricación y haciendo optima la distribución del espacio y por ende el tamaño del circuito desarrollado.

Para el desarrollo del prototipo se diseñaron dos circuitos impresos. El primer circuito es la placa base donde se realizan todas las conexión a la Raspberry Pi por medio de un cable *ribbon* de 40 pines, en esta placa se ubican un regulador de voltaje, el reloj de tiempo real, el conversor análogo digital, y una serie de borneas para la alimentación y las entradas análogas y digitales. En la Figura 67, se observa el diseño de la placa base.

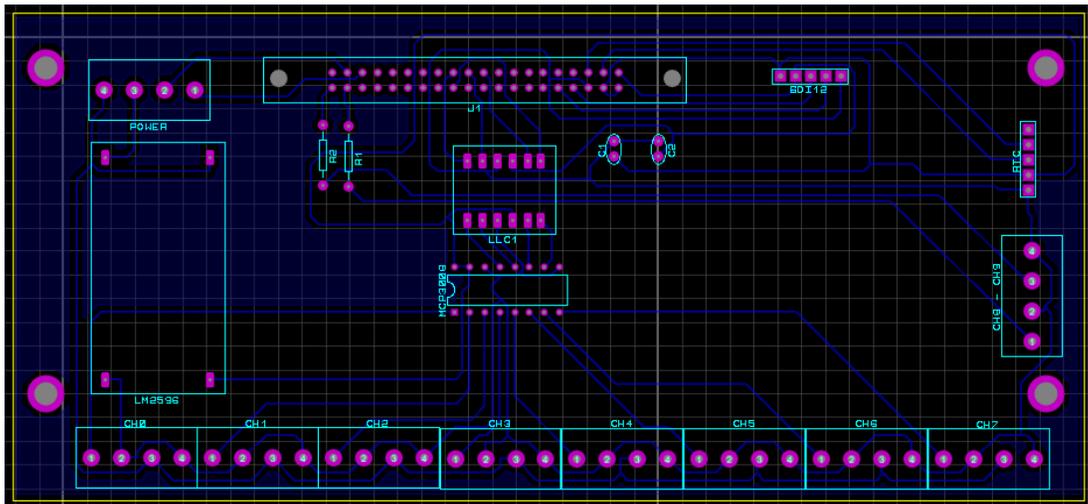


Figura 67. Diseño circuito placa base

En la Figura 68 se presentan imágenes de la placa base construida, se observa el estado de la tarjeta inicial sin componentes y la placa base con todos los componentes integrados en sus vistas frontal y posterior respectivamente.

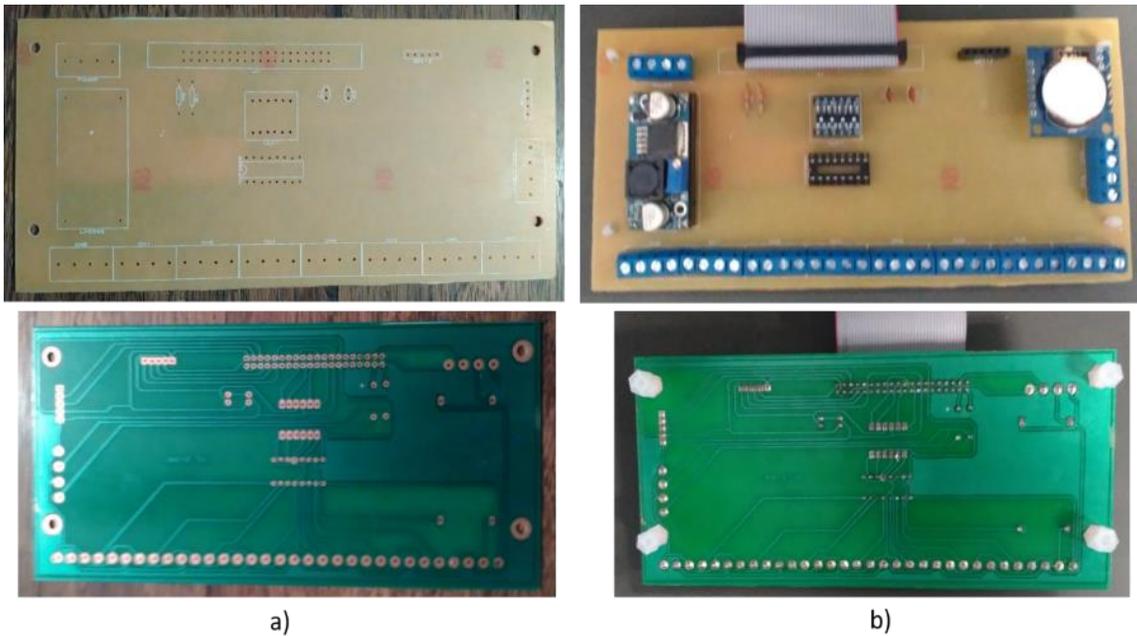


Figura 68. Fabricación placa base  
 a) Estado Inicial, b) Estado Final

El segundo circuito impreso se diseñó como un módulo independiente donde se ubica el microcontrolador Arduino utilizado para la entrada serial SDI-12, el cual se conecta físicamente a la Raspberry Pi por medio de un cable USB. En la Figura 69 se observa el módulo utilizado para la entrada del sensor SDI-12

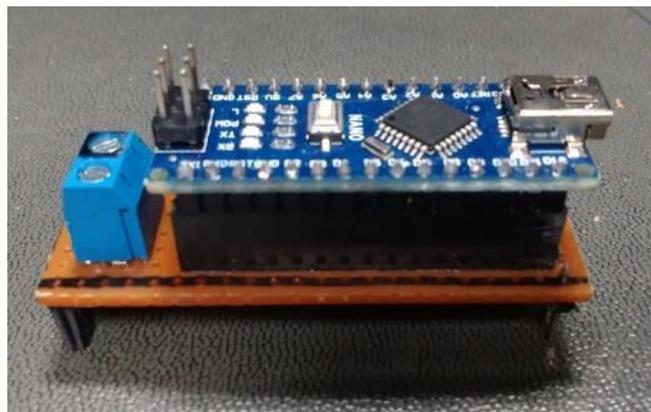


Figura 69. Módulo entrada SDI-12

El módulo de entrada SDI-12 se diseñó y construyó de tal forma que pudiera ser acoplado a la placa base, permitiendo tener todas las entradas integradas para la fácil conexión de

los sensores. En Figura 70, se observa el resultado final del circuito desarrollado e implementado para el prototipo.



Figura 70. Circuito final prototipo

## 6.2 Resultados de la medición de variables hidrometeorológicas

Para la verificación del funcionamiento del prototipo implementado, el IDEAM facilitó 4 sensores especializados, con el fin de obtener mediciones de variables como: temperatura del aire, humedad del aire, presión atmosférica, precipitación, dirección del viento, y velocidad del viento, los cuales son mostrados en la Figura 71, y cuya descripción detallada se realizó en la sección 3.6.

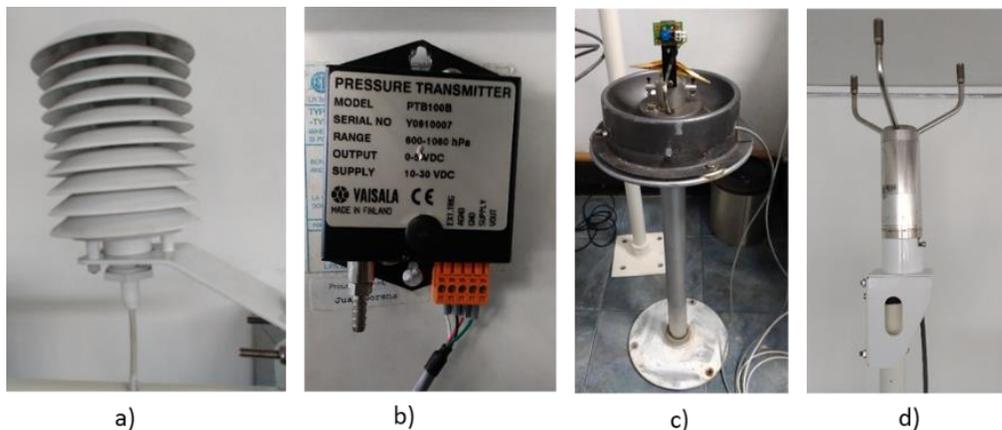


Figura 71. Sensores hidrometeorológicos utilizados

a) Temperatura y humedad, b) Presión, c) Precipitación, d) Dirección y velocidad del viento.

Por medio de la interfaz gráfica se, ingresan los datos de la configuración inicial correspondientes al nombre de la estación, código de la estación y los valores del tiempo de muestreo y tiempo de visualización, como es mostrado en la Figura 72, donde se observa que se realiza medición de las variables cada 2 minutos y donde además se activa el almacenamiento en la base de datos y la transmisión de los mismos al servidor remoto.

Figura 72. Configuración inicial para la verificación de funcionamiento

De igual forma se ingresaron los parámetros correspondientes a cada uno de los sensores, obteniendo la configuración mostrada en la Figura 73, donde se evidencia el uso de los tres tipos de sensores con que el prototipo fue diseñado para funcionar: sensores análogos, sensores digitales por pulsos, y sensores seriales con protocolo SDI12.

Sensores Analogos								
	Nombre	ID	Unidad	Decimale	In Min	In Max	Out Min	Out Max
<input checked="" type="checkbox"/>	Chanel 00	Temperatura del Aire	68	C	1	0	1	-40 60
<input checked="" type="checkbox"/>	Chanel 01	Humedad Del Aire	27	%	1	0	1	0 100
<input checked="" type="checkbox"/>	Chanel 02	Presion Atmosferica	255	HPa	1	0	5	600 1060

Sensor Digital						
	Nombre	ID	Unidad	Decimales	Resolucion	
<input checked="" type="checkbox"/>	Chanel 08	Precipitacion	240	mm	1	0.1

Sensor SDI-12							
	Nombre	ID	Unidad	Decimales	Esclavo	Posicion	
<input checked="" type="checkbox"/>	Chanel SDI12	Velocidad del Viento	103	m/s	0	0	1
<input checked="" type="checkbox"/>	Chanel SDI12	Direccion del Viento	104	gr	1	0	2

Figura 73. Configuración de los sensores utilizados para la verificación de funcionamiento.

Utilizando el sensor **Rotronic HC2**, en la entrada análoga “**chanel 0**”, se obtienen los resultados de la adquisición de datos para la variable de temperatura del aire por un

periodo de tiempo de 2 horas y 30 minutos los cuales con mostrados en la Figura 74, donde se observan valores en el rango de 22 a 24 °C.

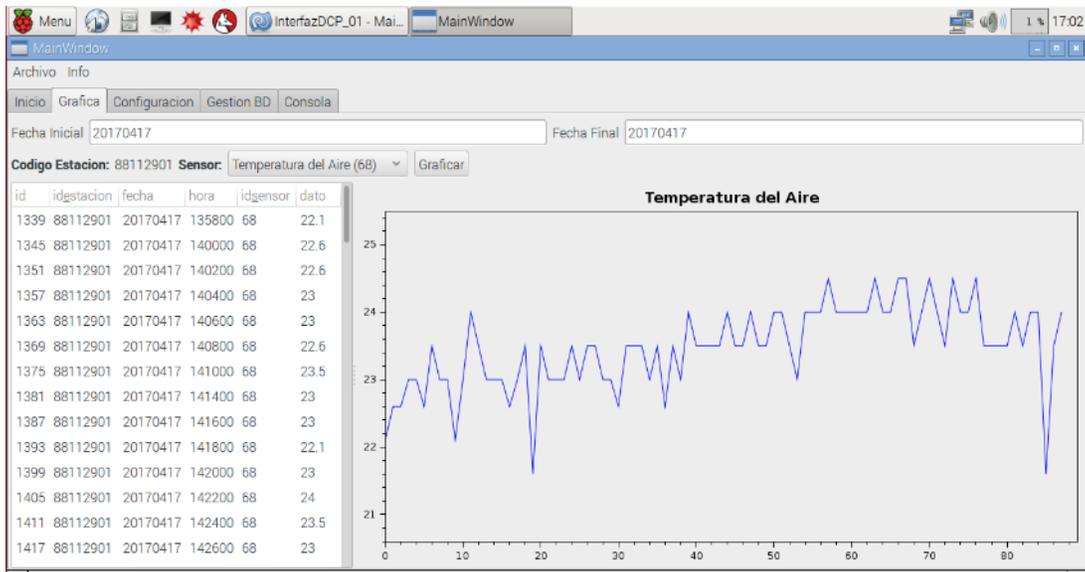


Figura 74. Adquisición de datos de temperatura con sensor ROTRONIC HygroClip HC2

Para la variable de humedad relativa del aire del sensor **Rotronic HC2**, se utiliza la entrada análoga “canal 1”, obteniendo los resultados de la adquisición de datos mostrados en la Figura 75, donde se observan valores entre 40 y 43% de humedad.

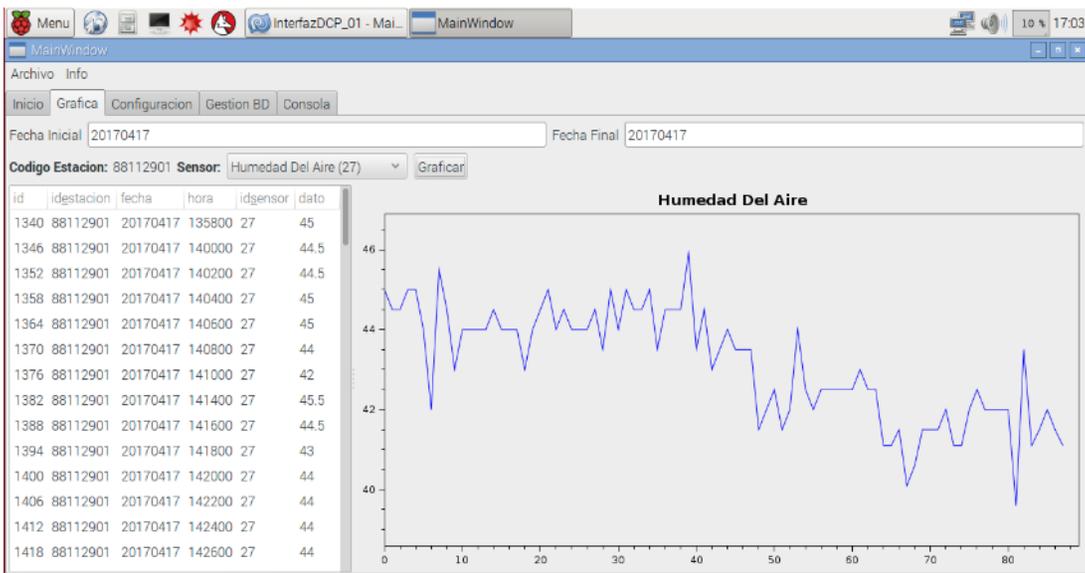


Figura 75. Adquisición de datos de humedad con sensor ROTRONIC HygroClip HC2

Utilizando el sensor **Vaisala PB100B**, y la entrada análoga “**chanel 2**”, se obtienen los resultados de la adquisición de datos para la variable de presión atmosférica, en un periodo de 2 horas los cuales son mostrados en la Figura 76.

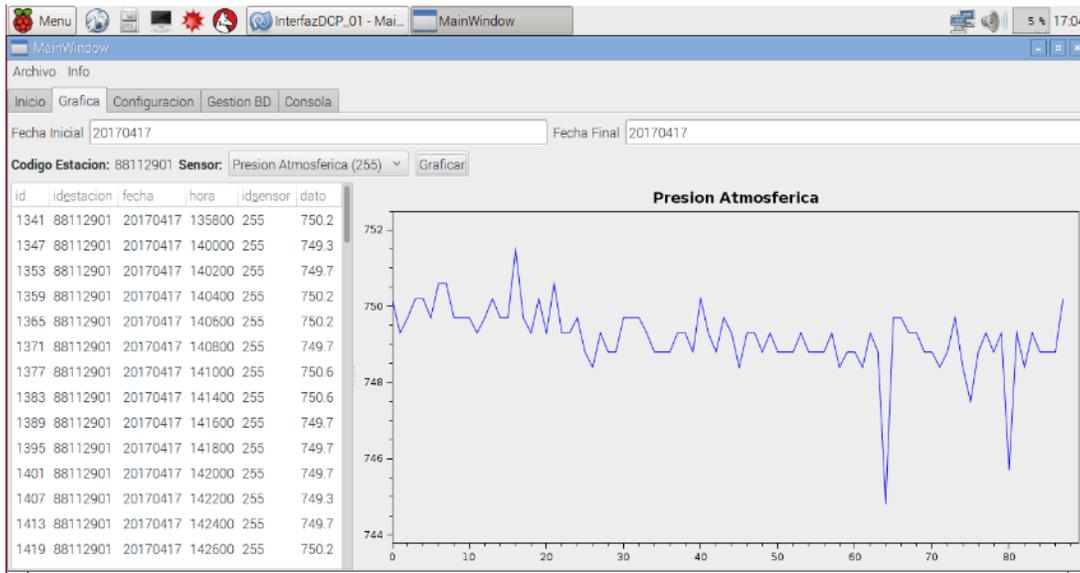


Figura 76. Adquisición de datos de presión atmosférica con sensor VAISALA PTB100B

Utilizando la entrada digital “**chanel 8**”, se conectó el sensor de precipitación **RIMCO 8500**, y se realiza la adquisición de datos, accionando manualmente el mecanismo de cazoletas del sensor obteniendo los valores mostrados en la Figura 77.

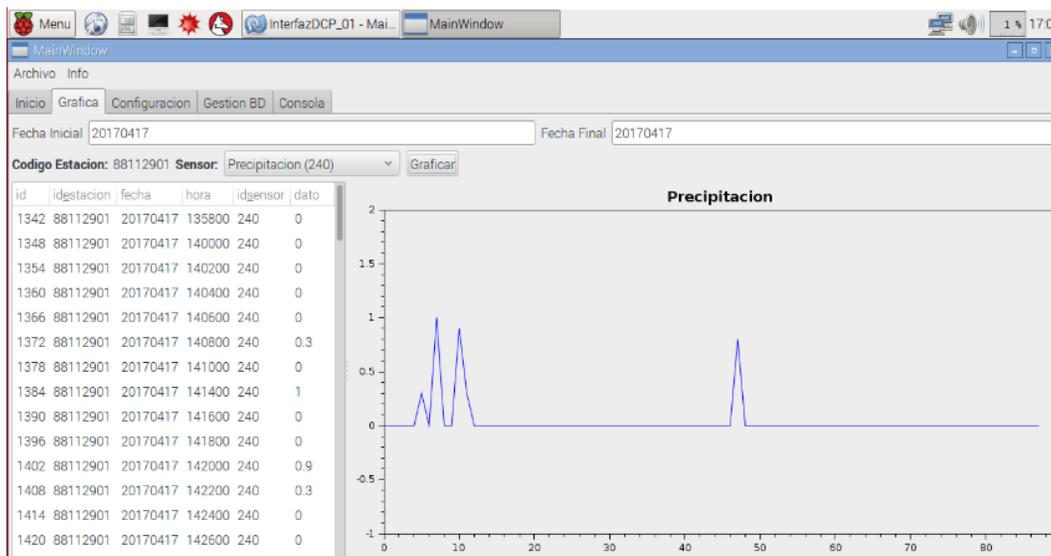


Figura 77. Adquisición de datos de precipitación con sensor RIMCO 8500

Utilizando la entrada “**chanel SDI-12**” se conecta el sensor **Vaisala WMT702**, el cual se configuro con el **esclavo 0**, y de acuerdo a las especificaciones técnicas, se realiza la medición de la velocidad el viento utilizando la **posición 1**, obteniendo los resultados de la adquisición de datos mostrados en la Figura 78.

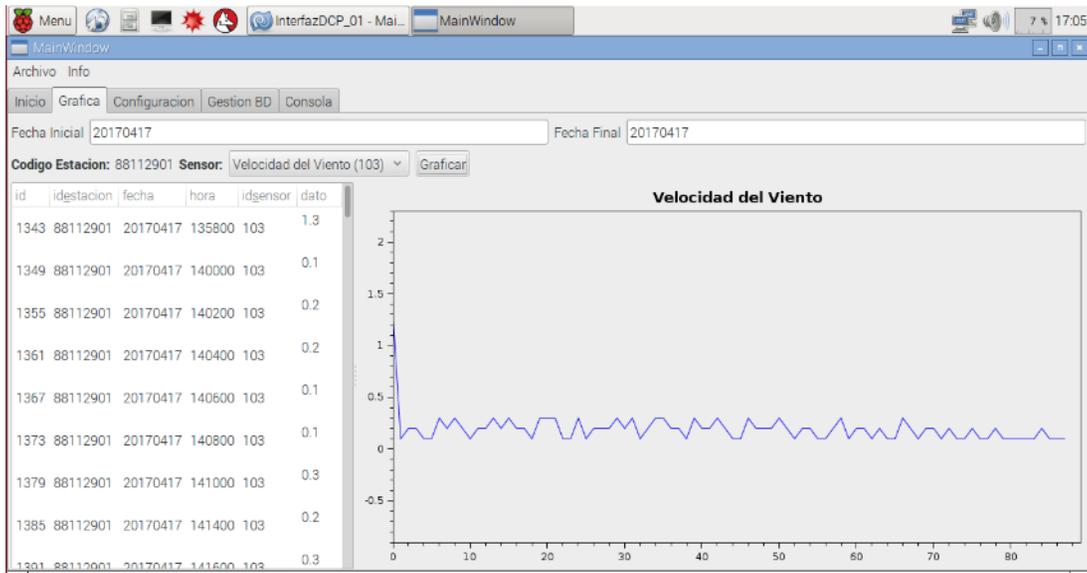


Figura 78. Adquisición de datos velocidad del viento con sensor VAISALA WMT702

Utilizando la **posición 2**, del mismo sensor, se obtienen los resultados de la adquisición de datos para la variable de dirección del viento los cuales son mostrados en la Figura 79.

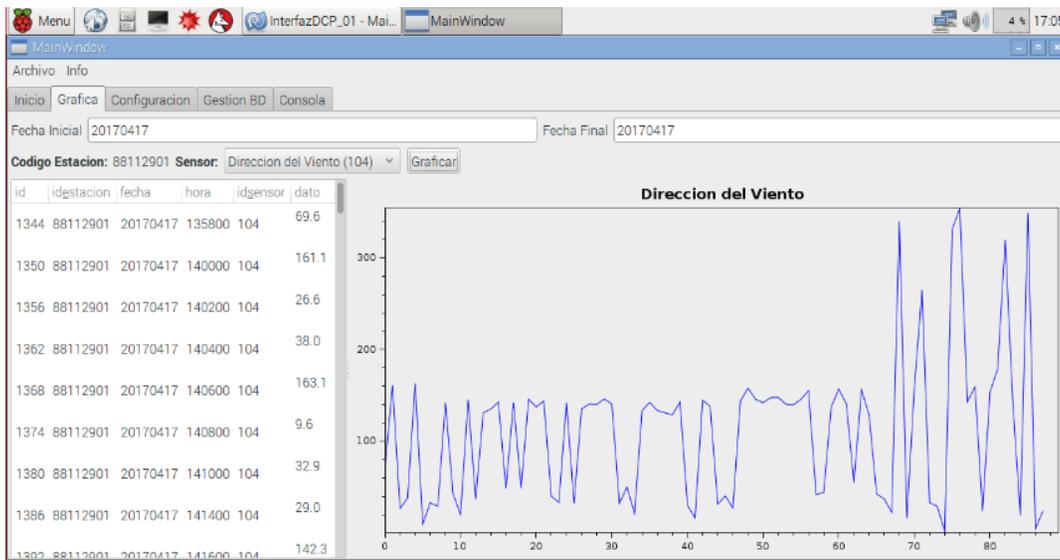
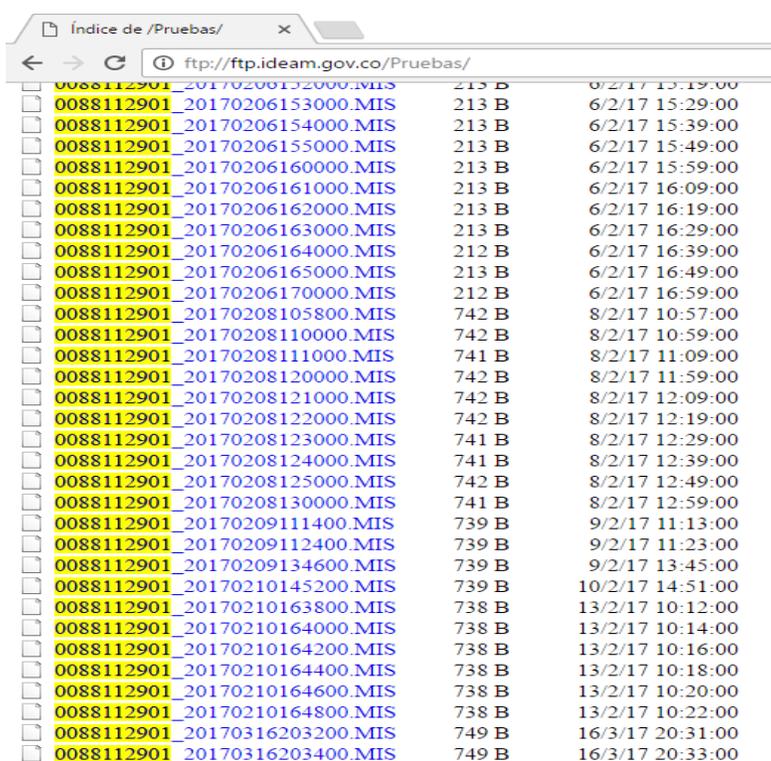


Figura 79. Adquisición de datos dirección de viento con sensor VAISALA WMT702

### 6.3 Integración de datos al sistema de visualización meteorológico Hydras3

Una vez implementado el sistema de transmisión, se realiza la verificación de datos en el software de visualización Hydras3, para lo cual es necesario comprobar la recepción del archivo de datos .MIS en el servidor FTP del IDEAM, ingresando a dicho servidor con el usuario y contraseña entregados por el IDEAM se puede observar como en la carpeta de “Pruebas” del servidor se almacenan los diferentes archivos, los cuales llegan según la periodicidad programada y con el código de estación que se ha seleccionado tal como se muestra en la Figura 80.



File Name	Size	Upload Date/Time
0088112901_20170206152000.MIS	213 B	6/2/17 15:19:00
0088112901_20170206153000.MIS	213 B	6/2/17 15:29:00
0088112901_20170206154000.MIS	213 B	6/2/17 15:39:00
0088112901_20170206155000.MIS	213 B	6/2/17 15:49:00
0088112901_20170206160000.MIS	213 B	6/2/17 15:59:00
0088112901_20170206161000.MIS	213 B	6/2/17 16:09:00
0088112901_20170206162000.MIS	213 B	6/2/17 16:19:00
0088112901_20170206163000.MIS	213 B	6/2/17 16:29:00
0088112901_20170206164000.MIS	212 B	6/2/17 16:39:00
0088112901_20170206165000.MIS	213 B	6/2/17 16:49:00
0088112901_20170206170000.MIS	212 B	6/2/17 16:59:00
0088112901_20170208105800.MIS	742 B	8/2/17 10:57:00
0088112901_20170208110000.MIS	742 B	8/2/17 10:59:00
0088112901_20170208111000.MIS	741 B	8/2/17 11:09:00
0088112901_20170208120000.MIS	742 B	8/2/17 11:59:00
0088112901_20170208121000.MIS	742 B	8/2/17 12:09:00
0088112901_20170208122000.MIS	742 B	8/2/17 12:19:00
0088112901_20170208123000.MIS	741 B	8/2/17 12:29:00
0088112901_20170208124000.MIS	741 B	8/2/17 12:39:00
0088112901_20170208125000.MIS	742 B	8/2/17 12:49:00
0088112901_20170208130000.MIS	741 B	8/2/17 12:59:00
0088112901_20170209111400.MIS	739 B	9/2/17 11:13:00
0088112901_20170209112400.MIS	739 B	9/2/17 11:23:00
0088112901_20170209134600.MIS	739 B	9/2/17 13:45:00
0088112901_20170210145200.MIS	739 B	10/2/17 14:51:00
0088112901_20170210163800.MIS	738 B	13/2/17 10:12:00
0088112901_20170210164000.MIS	738 B	13/2/17 10:14:00
0088112901_20170210164200.MIS	738 B	13/2/17 10:16:00
0088112901_20170210164400.MIS	738 B	13/2/17 10:18:00
0088112901_20170210164600.MIS	738 B	13/2/17 10:20:00
0088112901_20170210164800.MIS	738 B	13/2/17 10:22:00
0088112901_20170316203200.MIS	749 B	16/3/17 20:31:00
0088112901_20170316203400.MIS	749 B	16/3/17 20:33:00

Figura 80. Archivos “.MIS” en el servidor FTP del IDEAM

De igual forma se verifica el contenido de cada uno de los archivos “.MIS” comprobando la estructura correspondiente, y los datos capturados por los sensores tal como se muestra en la Figura 81.



```
<STATION>0088112901</STATION><SENSOR>0068</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 203400
<STATION>0088112901</STATION><SENSOR>0027</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 203400
<STATION>0088112901</STATION><SENSOR>0255</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 203400
<STATION>0088112901</STATION><SENSOR>9000</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 203400
<STATION>0088112901</STATION><SENSOR>0240</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 9.1
<STATION>0088112901</STATION><SENSOR>0103</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 11.0
<STATION>0088112901</STATION><SENSOR>0104</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>
20170316;203400; 14.0
```

Figura 81. Estructura del archivo “.MIS” recibido en el servidor del IDEAM.

Automáticamente el software Hydras3 realiza la lectura de los archivos “.MIS” haciendo posible la visualización, evaluación, y análisis de la información tanto en su versión WEB como en su versión de escritorio.

### 6.3.1 Resultados de las pruebas de transmisión

Se consiguen resultados de la adquisición y transmisión de datos, las cuales fueron obtenidas en un periodo de funcionamiento de más de **48 horas continuas**, almacenando información de las variables **cada 2 minutos**, donde se guardaron **8687 registros** en la base de datos. De la Figura 82 a la Figura 87, se puede observar la comparación de las gráficas del comportamiento de cada una de las variables, en la parte superior de las imágenes se muestra los registros de la interfaz gráfica del prototipo desarrollado y en la parte inferior los datos recibidos en el software de visualización Hydras3.

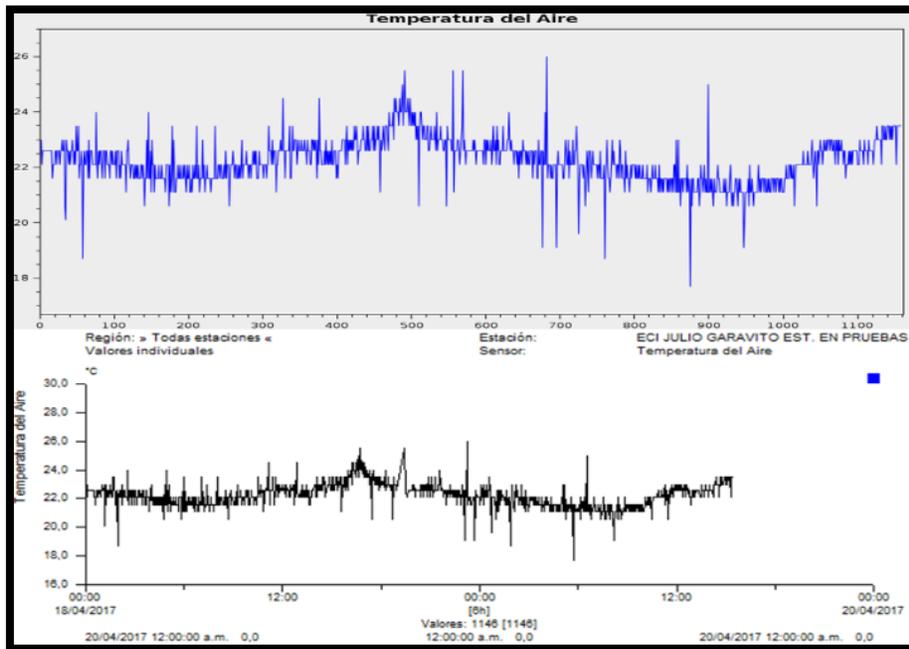


Figura 82. Resultados adquisición y transmisión de temperatura del aire  
a) Interfaz gráfica prototipo, b) Hydras3

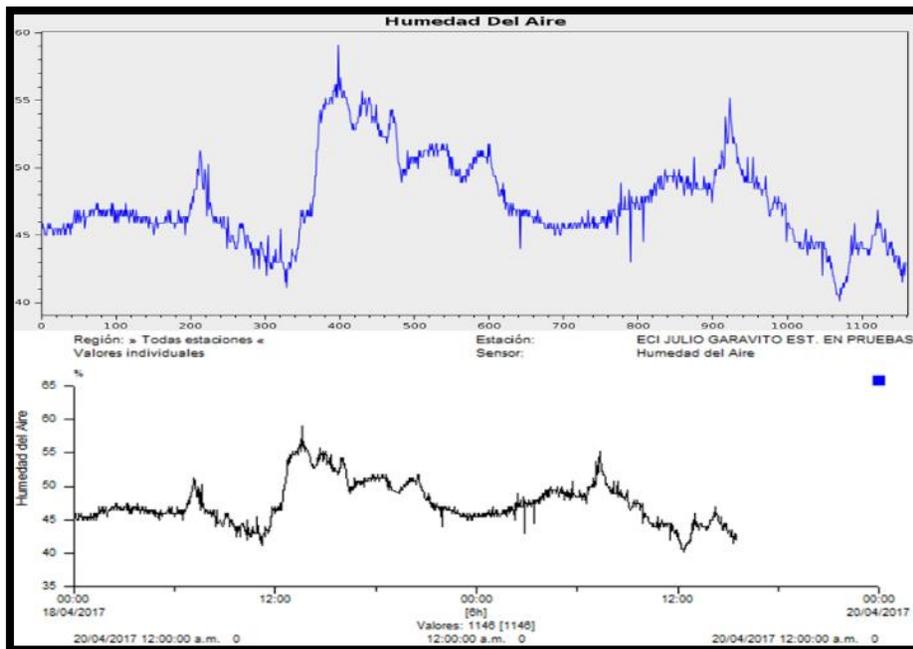


Figura 83. Resultados adquisición y transmisión de humedad del aire  
a) Interfaz gráfica prototipo, b) Hydras3

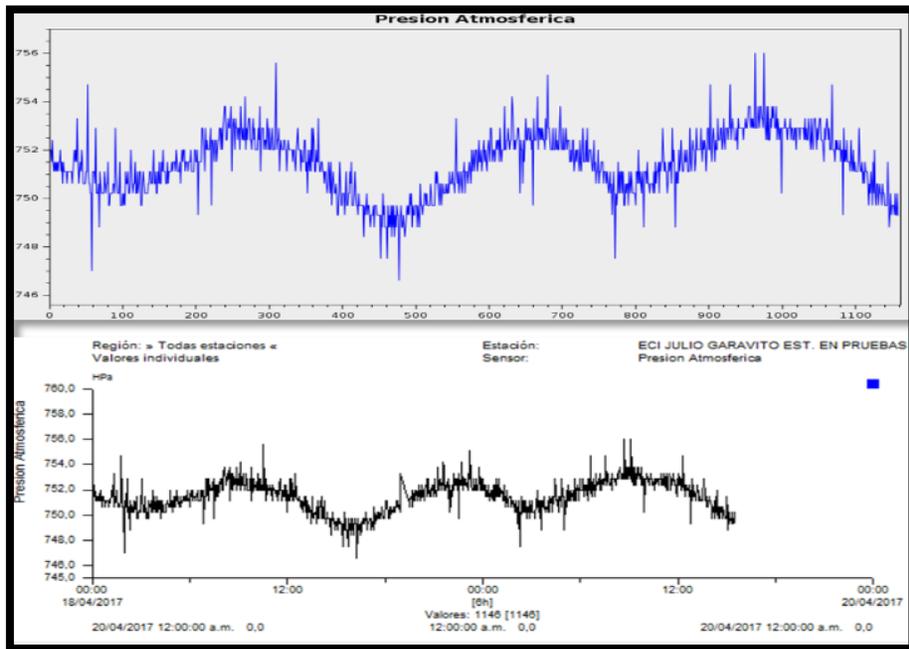


Figura 84. Resultados adquisición y transmisión de presión atmosférica  
a) Interfaz gráfica prototipo, b) Hydras3

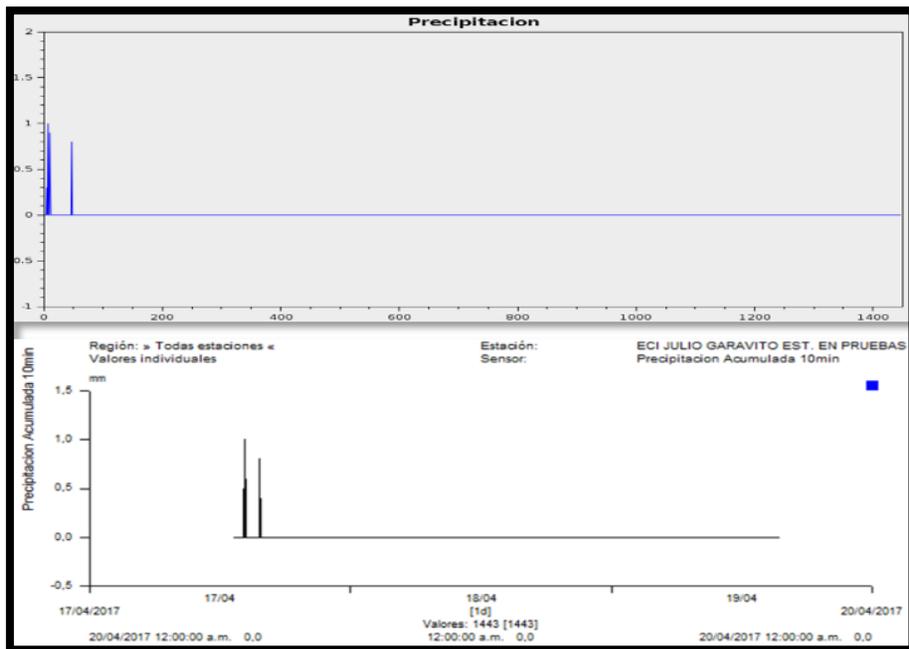


Figura 85. Resultados adquisición y transmisión de precipitación  
a) Interfaz gráfica prototipo, b) Hydras3

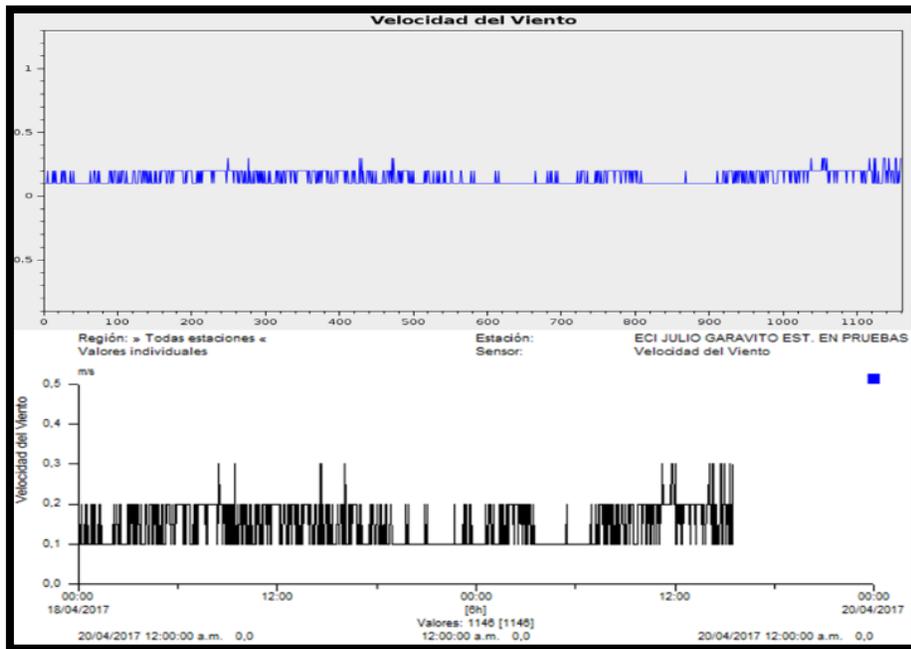


Figura 86. Resultados adquisición y transmisión de velocidad del viento  
 a) Interfaz gráfica prototipo, b) Hydras3

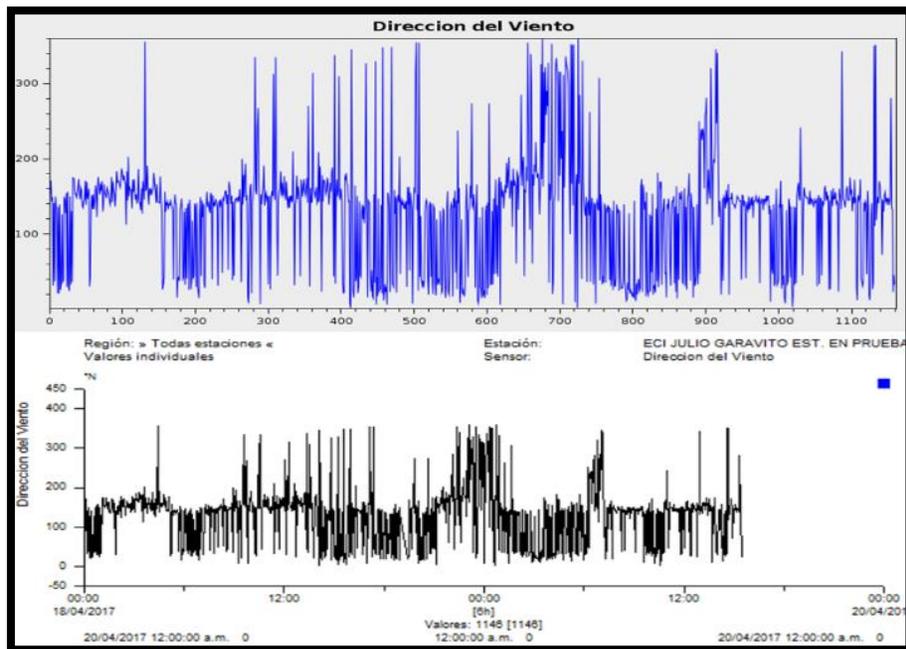


Figura 87. Resultados adquisición y transmisión de dirección del viento  
 a) Interfaz gráfica prototipo, b) Hydras3

#### 6.4 Montaje estructura física

Con el fin de proteger los dispositivos electrónicos que se usaron para la construcción del prototipo, ubicar los sensores y demás elementos utilizados, se implementa una estructura física básica la cual consta de un gabinete, un mástil principal, y un soporte para pluviómetro.

Se utilizó el gabinete **Campbell ENC10/12**, el cual está fabricado con un revestimiento de poliéster reforzado con fibra de vidrio, que lo hace apto para uso en intemperie, ya que cuenta con protección NEMA 4X<sup>4</sup>, (equivalente a IP66), protegiendo su interior contra polvo, lluvia, salpicaduras de agua y corrosión. El gabinete es utilizado para salvaguardar la tarjeta Raspberry Pi, la pantalla táctil, la placa base, el regulador de alimentación, y el sensor de presión atmosférica. En la Figura 88, se observa el gabinete utilizado.



Figura 88. Gabinete Campbell ENC10/12

Con base en el gabinete utilizado se realiza el diseño de una estructura metálica compuesta por una base atornillarle a la superficie y un mástil principal en forma de cruz, el cual sostiene el gabinete y los sensores de dirección y velocidad del viento, temperatura y humedad del aire así como la antena GPRS externa. Adicionalmente se

---

<sup>4</sup> NEMA 4X es el estándar establecido por la Asociación Nacional de Fabricantes Eléctricos para equipos de instalación en exterior o trabajo en ambientes hostiles con un grado de protección contra: Polvo llevado por el viento y lluvia, Salpicadura de agua, Agua directa de la manguera, Daño por la formación exterior de hielo, Resistente a la corrosión.

utiliza un soporte para la ubicación del pluviómetro. En la Figura 89, se observa el diseño de la estructura física y las dimensiones correspondientes.

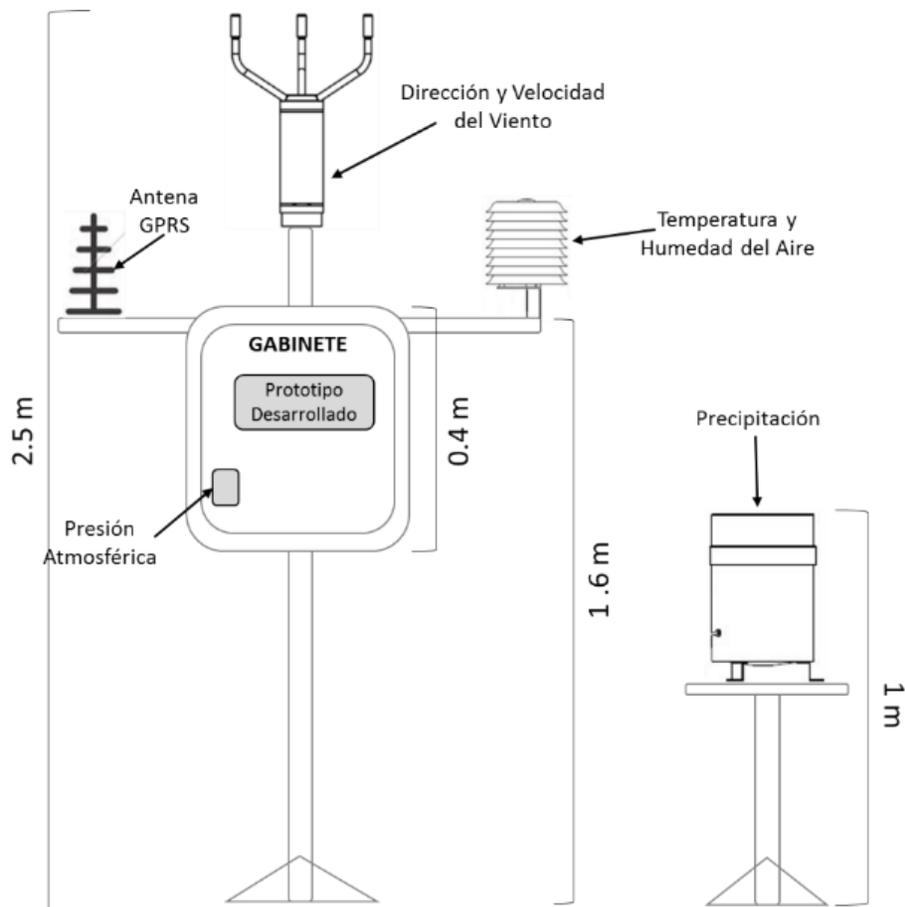


Figura 89. Diseño estructura física.

Todos los elementos utilizados para la construcción de la estructura física, fueron facilitados por el IDEAM, eran componentes que se encontraban almacenados y que después de algunas modificaciones y una capa de pintura se adaptaron perfectamente para la construcción del prototipo final. En la Figura 90 se puede observar los elementos utilizados en la estructura física, su estado inicial y su estado final antes de su ensamble.



Figura 90. Elementos utilizados en la estructura física  
 a) Mástil principal, b) Gabinete, c) Base de mástil

Luego de realizar el ensamble de los elementos utilizados se obtiene la estructura física final, que integra los sensores y el prototipo de plataforma colectora de datos desarrollada. En la Figura 91 y en la Figura 92 se puede observar el resultado final de la implementación.



Figura 91. Estructura física estación meteorológica

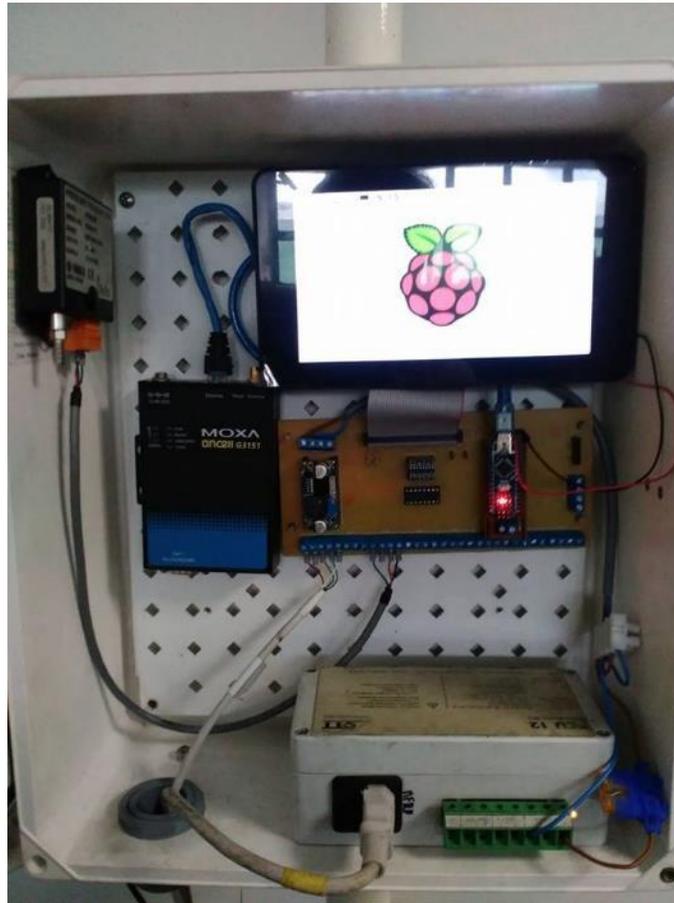


Figura 92. Interior del gabinete

## 6.5 Costos de fabricación

A continuación se presenta un análisis tentativo de los costos de fabricación del prototipo DCP, basado en los resultados obtenidos tras la finalización de las pruebas de funcionamiento. El presente estudio se realizó teniendo en cuenta únicamente los elementos principales involucrados durante el desarrollo del proyecto, para la producción de una (1) unidad, obteniendo así, el costo de fabricación unitario del prototipo.

Es importante mencionar que los resultados obtenidos únicamente se realizaron con el fin de obtener un prototipo funcional de una plataforma colectora de datos y no se contemplaron costos asociados a la operación del mismo, como por ejemplo, la fabricación de la estructura física, el gabinete de protección, el valor de instalación, el plan

internet para la transmisión de datos o los diferentes sensores hidrometeorológicos a utilizar.

Para el cálculo del costo de fabricación se deben tener en cuenta conceptos básicos de contabilidad de costos como lo son **materia prima, costos indirectos y mano de obra directa**, en vista que el prototipo DCP, es un producto tecnológico, con un componente de innovación de producto en la cual se obtiene un producto final realizando la combinación de conocimientos y tecnologías existentes, se considera **materia prima** todos los componentes electrónicos principales, los cuales se adquieren de proveedores externos y son la base para el desarrollo del producto final, como **costos indirectos**, se tiene en cuenta los componentes menores, elementos consumibles y la mano de obra indirecta en la fabricación del circuito impreso.

Finalmente como **mano de obra directa**, se reconoce una remuneración económica al encargado del diseño, programación, fabricación, y ejecución de pruebas al prototipo, quien debe cumplir con el perfil un profesional en ingeniera electrónica con experiencia profesional no inferior a tres (3) años, y que de acuerdo al “*Manual de Referencia de Tarifas para la Contratación de Servicios Profesionales de Ingeniería en Colombia*” [55] se encuentra en un escalafón 6 y su remuneración mínima debe ser de 6 salarios mínimos mensuales legales vigentes (SMMLV) lo que equivale aproximadamente a COP \$4.426.302, en vista que para la construcción de un dispositivo se necesitan aproximadamente 48 horas de trabajo se asume un valor horario de COP \$18.443 por lo cual el costo unitario de la mano de obra directa es de COP \$885.264.

En la Tabla 17, se relacionan los costos unitarios del prototipo DCP, obteniendo costos parciales de los componentes de materia prima por un valor de COP \$1.020.000, costo de los materiales indirectos por un valor de COP \$160.000 y costo de mano de obra directa por un valor de COP \$885.264, obteniendo bajo los supuestos realizados un valor total de fabricación de COP \$2.065.264.

<b>COSTOS UNITARIOS DE FABRICACION</b>					
	<b>ITEM</b>	<b>UNIDAD</b>	<b>CANTIDAD</b>	<b>COSTO UNIDAD</b>	<b>COSTO TOTAL</b>
<b>Materia Prima</b>	Raspberry Pi 2 Model B	Un	1	\$180.000	\$180.000
	Módulo Cámara Raspberry	Un	1	\$90.000	\$90.000
	Convertor SDI12 (Arduino Nano)	Un	1	\$80.000	\$80.000
	Tarjeta MicroSD 16 GB	Un	1	\$40.000	\$40.000
	Reloj en tiempo real - RTC	Un	1	\$10.000	\$10.000
	CAD - MCP3008	Un	1	\$20.000	\$20.000
	Módem GPRS + Antena	Un	1	\$250.000	\$250.000
	Pantalla Táctil + Carcasa	Un	1	\$350.000	\$350.000
<b>VALOR PARCIAL</b>					<b>\$1.020.000</b>
<b>Costos Indirectos</b>	Cables	Paquete	1	\$10.000	\$10.000
	Componentes Electrónicos(Resistencias, Condensadores, Soldadura, Conectores)	Un	1	\$50.000	\$50.000
	Fabricación Circuito Impreso	Un	1	\$100.000	\$100.000
<b>VALOR PARCIAL</b>					<b>\$160.000</b>
<b>Mano de Obra</b>	Profesional Ingeniería Electrónica	Hora	48	\$18.443	\$885.264
	<b>VALOR PARCIAL</b>				
<b>COSTO TOTAL DE FABRICACION</b>					<b>\$2.065.264</b>

Tabla 17. Costos de fabricación unitaria del prototipo DCP.

El costo de fabricación obtenido (COP\$ 2.065.264) se considera muy bajo con respecto al precio de venta de las plataformas colectoras de datos más reconocidas en el mercado como las mostradas anteriormente en la sección 3.5. El precio de venta del prototipo desarrollado debe estar ajustado a las tendencias, especificaciones técnicas y valores actuales del mercado.

En la Tabla 18 se presentan las principales características de algunas plataformas colectoras de datos junto con las especificaciones del prototipo desarrollado, de donde se puede discurrir la posibilidad de ingresar un producto competitivo en el mercado actual, ya que al contar con un costo de fabricación tan bajo, se puede considerar realizar una mayor inversión económica para mejorar ciertas características técnicas como la

resolución y exactitud del convertor análogo digital y las especificaciones físicas de robustez, para así, como trabajo futuro plantear un modelo de negocio rentable con un margen de ganancia y producción sostenible.

	Prototipo DCP	SIAP + MICROS DA15K	SEBA Unilog	SUTRON 8310	OTT NetDL 500/1000
<b>Sistema Operativo</b>	Raspbian	Windows CE 6.0	NA	NA	NA
<b>Memoria</b>	16GB – MicroSD	512MB	4 MB	65MB	256MB
<b>RAM</b>	1GB	64MB	NA	32MB	4MB
<b>Display</b>	Touch - Screen 7"	Touch - Screen 3.5"	LCD 3x16	LCD 2x20	LCD 4x20
<b>Entradas Análogas</b>	8 - (0-1V, 0-2.5V, 0-5V)	4x Diferencial: - 2.5 – 2.5V 8x Single ended: 0 – 2.5V, 0– 20mA, 4–20mA 4x Pt100	8 entradas bi/Unipolares (0-1 V, 0-20 mA, 6-20 mA, 0-5V, configurable)	8 (0.1 a 5V)	4x Impulse/Frequency/Status 8x analogue input (0-50mV, 0-5V, 0-20mA, 4-20mA, PT100)
<b>Resolución ADC</b>	10 Bit	24	16	16	24
<b>Entradas Digitales</b>	2	4	2	4	2
<b>Entradas SDI-12</b>	1	2	1	2	1
<b>Puertos USB</b>	3	3	1	1	2
<b>Interfaces</b>	Ethernet	1x Ethernet 10/100 Base T 4x RS232 / RS485	RS232, RS485, LAN	RS-232, RS-485, LAN	2 x RS-232 (full DB9) LAN 10Mbit, 10BaseT RS485
<b>Módem GPRS</b>	Externo	Integrado	Externo	Integrado	Integrado
<b>Valor Aprox</b>	<b>\$2.065.264**</b>	<b>\$22.000.000</b>	<b>\$14.000.000</b>	<b>\$11.000.000</b>	<b>\$15.000.000</b>

Tabla 18. Comparación de especificaciones técnicas y costos de plataformas colectoras de datos con el prototipo DCP.

\*\*Costo de fabricación

## 7 CONCLUSIONES Y RECOMENDACIONES

Se diseñó e implemento un prototipo basado en un computador de placa reducida, capaz de adquirir y visualizar datos hidrometeorológicos de sensores electrónicos especializados, almacenando la información en una base de datos relacional, realizando él envío periódico de las variables al software de visualización nacional del **IDEAM Hydras3**. Adicionalmente integra un sistema de registro fotográfico que permite capturar imágenes y guardarlas en la memoria para su posterior análisis.

El sistema es capaz de realizar la captura de información de hasta 8 sensores de tipo análogo entre 0 y 5V, 2 sensores digitales con salidas en pulsos, y 1 entrada para sensores seriales con protocolo SDI-12. El prototipo funcionó adecuadamente según los objetivos planteados, igualando las funcionalidades de las plataformas colectoras de datos más utilizadas actualmente en el mercado. Sin embargo se deben realizar de pruebas para comprobar el comportamiento el prototipo expuesto a condiciones de campo y la calidad de las mediciones capturadas.

El utilizar un computador de placa reducida como la Raspberry Pi, es de gran utilidad, debido a sus características de procesamiento multitarea, capacidad de ejecutar sistemas operativos basados en Linux y uso de las diferentes interfaces de conexión con dispositivos externos, con lo cual se desarrollan aplicaciones para la lectura, procesamiento y visualización de la información adquirida por los sensores.

Se utiliza una base de datos relacional desarrollada con MySql la cual es de gran ayuda ya que cuenta con complementos gratuitos para la integración con herramientas como Python y MonoDevelop y adicionalmente permite almacenar gran cantidad de información ordenando los datos en tablas permitiendo realizar consultas y exportaciones de datos a archivos de texto plano.

Se realizaron pruebas de funcionamiento con tres (3) sensores análogos de temperatura del aire, humedad del aire y presión atmosférica, un (1) sensor digital por pulsos de precipitación, y un (1) sensor serial con protocolo SDI-12 de dirección y velocidad del viento. Adicionalmente el sistema estaría en capacidad de adquirir información de otro tipo

de sensores diferentes a variables hidrometeorológicas, como pueden ser sensores de tipo ambiental o industrial. En la Tabla 19 se relacionan diferentes tipos de sensores que pueden ser integrados al prototipo desarrollado.

Sensores Análogos	Sensores Digitales	Sensores SDI-12*
<ul style="list-style-type: none"> <li>• Temperatura del Suelo</li> <li>• Humedad del Suelo</li> <li>• Evaporacion</li> <li>• Proximidad</li> <li>• Luminosidad</li> <li>• Dioxido de Carbono (CO2)</li> <li>• Caudal</li> <li>• Distancia</li> </ul>	<ul style="list-style-type: none"> <li>• Precipitacion tipo Balancin</li> <li>• Velocidad del Viento</li> <li>• Fin de Curso</li> <li>• Contador de Revoluciones</li> </ul>	<ul style="list-style-type: none"> <li>• Direccion y Velocidad del viento</li> <li>• Nivel</li> <li>• Precipitacion Solida</li> <li>• Temperatura del aire</li> <li>• Humedad dela aire</li> <li>• PH</li> <li>• Conductancia</li> </ul> <p><b>* Cualquier sensor que cumpla con el Protocolo SDI12- V1.3</b></p>

Tabla 19. Diferentes tipos de sensores comerciales con la capacidad de ser integrados al prototipo desarrollado.

Se integró un sistema de captura de imágenes a la plataforma colectora de datos, convirtiéndose en un componente de innovación para el apoyo a las observaciones hidrometeorológicas, ya que en la actualidad, ninguna de las estaciones meteorológicas del IDEAM, cuenta con este tipo de sistemas fotográfico, lo cual puede servir como un registro alternativo para medición de variables como deshielo o nivel, así como también el monitoreo de diversos fenómenos atmosféricos.

Se desarrolló una aplicación gráfica en MonoDevelop utilizando el lenguaje de programación orientado a objetos C#, para la visualización de información y la configuración de los parámetros de adquisición y transmisión de datos obtenidos de los sensores, cada una de las entradas para los sensores se puede configurar de forma personalizada por medio de la interfaz, ingresando los valores a los parámetros correspondientes a de cada tipo de sensor, lo cual es de gran ayuda ya que a diferencia de otras plataformas colectoras que se encuentran en el mercado, esta aplicación no depende de un software externo para realizar cambios de configuración o análisis simples de verificación de funcionamiento.

Utilizando Python3 se crearon diferentes funciones para el manejo de los pines GPIO y el uso de los diferentes protocolos de la tarjeta Raspberry Pi con el fin de realizar la adquisición, procesamiento y transmisión de los datos provenientes de los sensores, para luego ser integrados a la interfaz gráfica de visualización desarrollada en MonoDevelop. Sin embargo, durante el desarrollo del trabajo se descubren algunas librerías de C# que pueden llegar a ser útiles para en el futuro realizar una única aplicación concentrando el manejo de pines GPIO, las interfaces de comunicación y la parte gráfica en una única aplicación en C# usando las librerías Raspberry# IO y MonoDevelop.

Se realizó la integración de los datos capturados por el prototipo al sistema de visualización nacional **Hydras3** utilizado por el **IDEAM**, se verificó la correcta recepción de los archivos de datos “.MIS” en el servidor FTP y la visualización de datos en el software con pruebas de 48 horas de funcionamiento continuo registrando valores de temperatura y humedad del aire, presión atmosférica, precipitación y dirección y velocidad del viento.

Las pruebas de transmisión de datos se realizaron utilizando la red celular GPRS en la ciudad de Bogotá, en la cual la calidad y fuerza de la señal siempre fue estable y no presento ningún tipo de pérdidas. Para poder determinar el grado de confianza y estabilidad del módem celular utilizado, se deberían realizar pruebas en ubicaciones remotas donde no se tenga buena cobertura celular, actividad que se consideraría en posibles trabajos futuros.

El prototipo cuenta con un conversor análogo digital con resolución de 10bits, con lo cual se pueden realizar mediciones de varios sensores análogos que se encuentran en el mercado, sin embargo de acuerdo a los resultados obtenidos en las pruebas de funcionamiento, como trabajo futuro y con el fin de mejorar el funcionamiento del prototipo, se recomienda utilizar un conversor análogo digital de mayor resolución y exactitud, para realizar capturas de sensores que manejan salidas de voltaje en el orden de los microvoltios ( $10^{-9}$  V), como los sensores de radiación global o radiación UVB.

Se fabricó una placa base con el circuito implementado para la lectura de señales análogas, digitales y seriales con protocolo SDI-12, con el cual se pudo verificar positivamente el funcionamiento del diseño planteado. Para un trabajo futuro se puede

mejorar el circuito impreso utilizando componentes de montaje superficial y trazado multicapa así como también el resguardo de los elementos dentro de una carcasa con un alto grado de protección con certificación NEMA o IP.

Basados en los cálculos realizados para el dimensionamiento del sistema de alimentación, se concluye que es posible alimentar el prototipo desarrollado por medio de un sistema solar fotovoltaico, sin embargo el consumo de potencia es mayor al de otras plataformas colectoras de datos que se encuentran en el mercado, lo cual implica utilizar paneles y baterías de mayor tamaño y costo. Como trabajo futuro se puede plantear el mejoramiento del consumo del prototipo utilizando las nuevas versiones de la tarjeta Raspberry Pi, cambiando la pantalla de 7" por una de menor tamaño y el apagado y encendido periódico del módem celular.

Como trabajo futuro, se plantea la posibilidad de implementar una aplicación web que permita tener funcionalidades similares a **Hydras3**, a fin de no depender de un software propietario adicional para realizar las tareas de visualización y análisis de datos hidrometeorológicos, lo que sería de gran utilidad para la comunidad cercana al análisis de datos hidrometeorológicos.

## 8 REFERENCIAS

- [1] IDEAM, «Catalogo Nacional de Estaciones IDEAM». Actualización de Enero-2016.
- [2] R. S. Pressman, *Ingeniería del software: un enfoque práctico*. México: McGraw-Hill, 2010.
- [3] «MODELO CASCADA Y MODELO MSF», *calameo.com*. [En línea]. Disponible en: <http://www.calameo.com/read/00035919610a2f1a84f70>. [Accedido: 25-mar-2018].
- [4] S. Nuñez y E. Gentile, *La Observación Meteorológica*. Universidad de Buenos Aires, 2004.
- [5] (OMM) Organización Meteorológica Mundial, «OMM en pocas palabras/OMM». [En línea]. Disponible en: [https://www.wmo.int/pages/about/index\\_es.html](https://www.wmo.int/pages/about/index_es.html). [Accedido: 24-mar-2016].
- [6] World Meteorological Organization (WMO), «Global Observing System (GOS)». [En línea]. Disponible en: <https://www.wmo.int/pages/prog/www/OSY/GOS.html>. [Accedido: 16-mar-2016].
- [7] World Meteorological Organization (WMO), *Manual on the WMO Integrated Global Observing System*. 2015.
- [8] World Meteorological Organization (WMO), «Estrategia de Desarrollo y Ejecución del WIGOS». 2012.
- [9] World Meteorological Organization (WMO), «WIGOS - Sistema mundial integrado de sistemas de observación de la OMM». 2011.
- [10] (OMM) Organización Meteorológica Mundial, *Manual del Sistema Mundial de Observación: volumen I - Aspectos mundiales*, vol. 1. Ginebra: OMM, 2013.
- [11] World Meteorological Organization (WMO), *Guide to Meteorological Instruments and Methods of Observation*. 2008.
- [12] (WMO) World Meteorological Organization, *Guide to Hydrological Practices, Volume I*. WMO, 2008.
- [13] (OMM) Organización Meteorológica Mundial, *Guía de prácticas climatológicas*. Ginebra: OMM, 2011.
- [14] (WMO) World Meteorological Organization, *International meteorological vocabulary*. WMO, 1992.

- [15] P. Battista, (WMO) World Meteorological Organization, (IATA) Institute of Agrometeorology and Environmental Analysis for Agriculture, y (CNR) Regional Meteorological Training Center, *Manual on instrumentation and operations for automatic weather stations for agrometeorological application*. WMO, 2000.
- [16] Boshell V J. F., «Gestión de información agroclimática en Colombia», Programa Adaptación al Cambio Climático en la Región Andina, Informe de Consultoría, 2012.
- [17] «Corporaciones Autónomas Regionales | Ministerio de Ambiente y Desarrollo Sostenible». [En línea]. Disponible en: <https://www.minambiente.gov.co/index.php/component/content/article?id=885:plantilla-a-areas-planeacion-y-seguimiento-33>. [Accedido: 06-may-2016].
- [18] CRED, EM-DAT, y UNISDR, «The Human Cost of Weather Related Disasters 1995-2015». 2015.
- [19] Munich Re y NatCatSERVICE |, «Loss Events Worldwide 1980 – 2014». ene-2015.
- [20] D. Guha-Sapir, P. Hoyois, y R. Below, «Annual Disaster Statistical Review 2014: The numbers and trends». CRED, 2015.
- [21] UNGRD, «Consolidado de Atención de Emergencias 2014». 2014.
- [22] «Tragedia de Armero», *Wikipedia, la enciclopedia libre*. 08-abr-2016.
- [23] C. E. E. P. El Pais, «Cifra de muertos por avalancha en Mocoa asciende a 314», *elpais.com.co*. [En línea]. Disponible en: <http://www.elpais.com.co/colombia/cifra-de-muertos-por-avalancha-en-mocoa-asciende-a-314.html>. [Accedido: 23-abr-2017].
- [24] C. E. E. Tiempo, «La avalancha que provocó la peor catástrofe en la historia de Mocoa», *El Tiempo*. [En línea]. Disponible en: <http://www.eltiempo.com/colombia/otras-ciudades/avalancha-en-mocoa-cifras-de-la-tragedia-y-causas-73984>. [Accedido: 23-abr-2017].
- [25] Group On Earth Observations, «Cómo las observaciones de la Tierra pueden apoyar el desarrollo sostenible en América Latina». 2011.
- [26] REN21, *Renewables 2015 Global Status Report*. 2015.
- [27] IRENA, «Renewable Energy Capacity Statistics 2015». 2015.
- [28] «Wind energy | Renewable energy | Statistical Review of World Energy 2015 | Energy economics | BP Global», *bp.com*. [En línea]. Disponible en: <http://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy/renewable-energy/wind-energy.html>. [Accedido: 18-abr-2016].

- [29] UPME, *Integración de las energías renovables no convencionales en Colombia*. 2015.
- [30] «OMS | Prevención del cáncer», WHO. [En línea]. Disponible en: <http://www.who.int/cancer/prevention/es/>. [Accedido: 17-abr-2016].
- [31] OMS, «Las radiaciones ultravioleta y la salud humana». 2009.
- [32] «OMS | Meningitis meningocócica», WHO. [En línea]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs141/es/>. [Accedido: 18-abr-2016].
- [33] Q. Xuewu y C. Tijing, «Design and implementation of data acquisition system based on Agilent 3458A», en *2013 IEEE 11th International Conference on Electronic Measurement Instruments (ICEMI)*, 2013, vol. 1, pp. 28-31.
- [34] M. Branzila, F. Mariut, y D. Petrisor, «Virtual instrument developed for adcon weather station», en *2012 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2012, pp. 853-856.
- [35] D. G. C. Niño, G. A. M. Vargas, y M. F. R. Perez, «Monitoring system of a weather station via IP», en *2010 IEEE ANDESCON*, 2010, pp. 1-6.
- [36] A. Shaout, Y. Li, M. Zhou, y S. Awad, «Low cost embedded weather station with intelligent system», en *Computer Engineering Conference (ICENCO), 2014 10th International*, 2014, pp. 100-106.
- [37] O. Krejcar, «Low cost weather station with remote control», en *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, 2012, pp. 463-468.
- [38] R. Lajara, J. Alberola, J. Pelegri, T. Sogorb, y J. V. Llario, «Ultra Low Power Wireless Weather Station», en *International Conference on Sensor Technologies and Applications, 2007. SensorComm 2007*, 2007, pp. 469-474.
- [39] M. Popa y C. Iapa, «Embedded weather station with remote wireless control», en *Telecommunications Forum (TEFOR), 2011 19th*, 2011, pp. 297-300.
- [40] R. C. Brito, F. Favarim, G. Calin, y E. Todt, «Development of a low cost weather station using free hardware and software», en *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, 2017, pp. 1-6.
- [41] R. Shete y S. Agrawal, «IoT based urban climate monitoring using Raspberry Pi», en *Communication and Signal Processing (ICCSP), 2016 International Conference on*, 2016, pp. 2008–2012.

- [42] T. Savić y M. Radonjić, «One approach to weather station design based on Raspberry Pi platform», en *Telecommunications Forum Telfor (TEFOR), 2015 23rd*, 2015, pp. 623-626.
- [43] P. Xie *et al.*, «Sensing wind for environmental and energy applications», en *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CICT 2014)*, 2014, pp. 64-69.
- [44] S. Jindarat y P. Wuttidittachotti, «Smart farm monitoring using Raspberry Pi and Arduino», en *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, 2015, pp. 284-288.
- [45] «OTT netDL Data Logger - OTT Hydromet USA». [En línea]. Disponible en: <http://www.ott.com/en-us/products/data-logging-and-telemetry-4/ott-netdl-data-logger-972/>. [Accedido: 15-mar-2018].
- [46] «CR6 - Measurement and Control Datalogger». [En línea]. Disponible en: <https://www.campbellsci.com/cr6>. [Accedido: 15-mar-2018].
- [47] «Sistemas de Adquisición de Datos - Siap+Micros». [En línea]. Disponible en: <http://www.siapmicros.com/es/prodotti/sistemas-de-adquisicion-de-datos/>. [Accedido: 15-mar-2018].
- [48] «Raspberry Pi 2 Model B», *Raspberry Pi*. [En línea]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Accedido: 06-may-2016].
- [49] SDI-12 Support Group, «SDI-12 A Serial-Digital Interface Standard for Microprocessor-Based Sensors Version 1.4». 10-ago-2016.
- [50] J. Coppock, «Development of a Raspberry Pi based, SDI-12 sensor environmental data logger», 2015.
- [51] Organización Meteorológica Mundial, *Atlas internacional de nubes. 1, Manual de observación de nubes y otros meteoros 1, Manual de observación de nubes y otros meteoros*. Ginebra (Suiza): Organización Meteorológica Mundial, 1993.
- [52] EURAMET, «Guidelines on the Calibration of Digital Multimeters». feb-2015.
- [53] Centro Español de Metrología, «PROCEDIMIENTO EL- 001 PARA LA CALIBRACIÓN DE MULTÍMETROS DIGITALES». 02-feb-2011.
- [54] «Fluke 725 - Calibrador de proceso de múltiples funciones | Fluke Espanol». [En línea]. Disponible en: <http://www.fluke.com/fluke/coes/calibradores-de->

procesos/calibradores-multifuncion/fluke-725.htm?pid=56117. [Accedido: 25-jul-2017].

- [55] Asociación Colombiana de Ingenieros, «Manual de Referencia de Tarifas para la Contratación de Servicios Profesionales de Ingeniería en Colombia». nov-2008.
- [56] J. A. Alonso Lorenzo, «Manual para instalaciones fotovoltaicas autónomas». SunFields Eurpe, 2015.
- [57] IDEAM, «Atlas de Radiacion Solar, Ultravioleta, y Ozono de Colombia», <http://atlas.ideam.gov.co/visorAtlasRadiacion.html>. [En línea]. Disponible en: <http://atlas.ideam.gov.co/visorAtlasRadiacion.html>.

## ANEXO A. CONFIGURACIÓN RELOJ EN TIEMPO REAL RTC-DS1307 POR I<sup>2</sup>C

A continuación se describe el procedimiento para realizar la configuración del módulo de reloj en tiempo real DS1307 con el fin de mantener la información de fecha y hora actualizadas y que sirva como respaldo cuando no se tenga una conexión a internet.

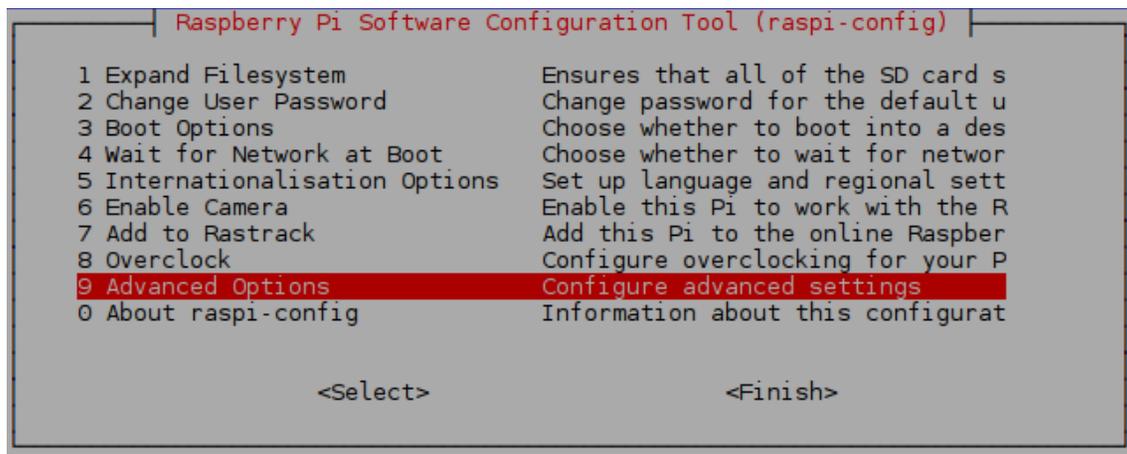
Inicialmente es necesario instalar los complementos para Python y el complemento de herramientas para lectura del protocolo i<sup>2</sup>c, para lo cual desde la terminal del sistema operativo se ingresan los comandos:

```
pi@raspberrypi:~ $ sudo apt-get install python-smbus
pi@raspberrypi:~ $ sudo apt-get install i2c-tools
```

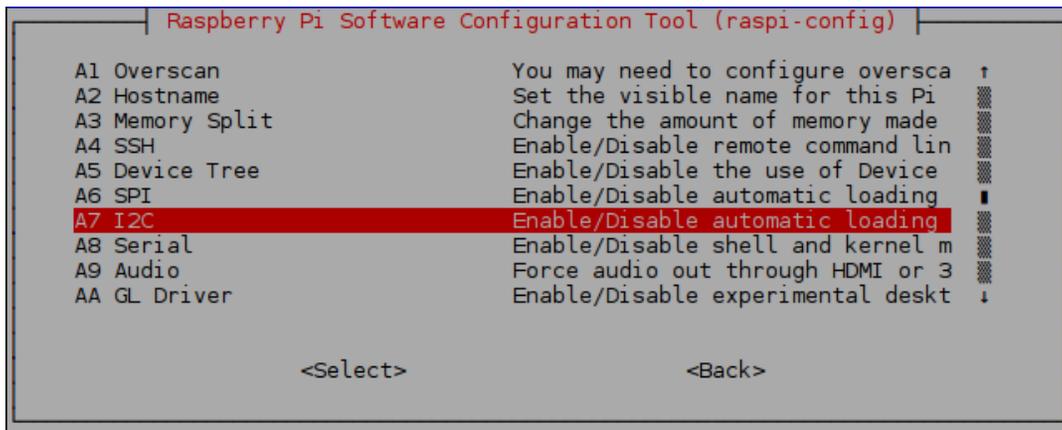
Luego de instalar los complementos se debe habilitar la interfaz i2c en el sistema operativo de la Raspberry Pi, para lo cual se ingresa a la configuración por medio del comando:

```
pi@raspberrypi:~ $ sudo raspi-config
```

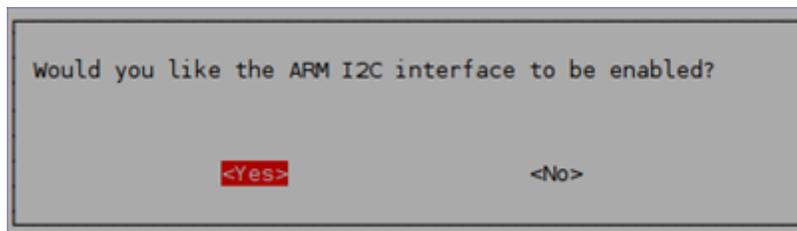
Luego entramos a la opción **9 Advanced Options** como se muestra en la siguiente imagen:



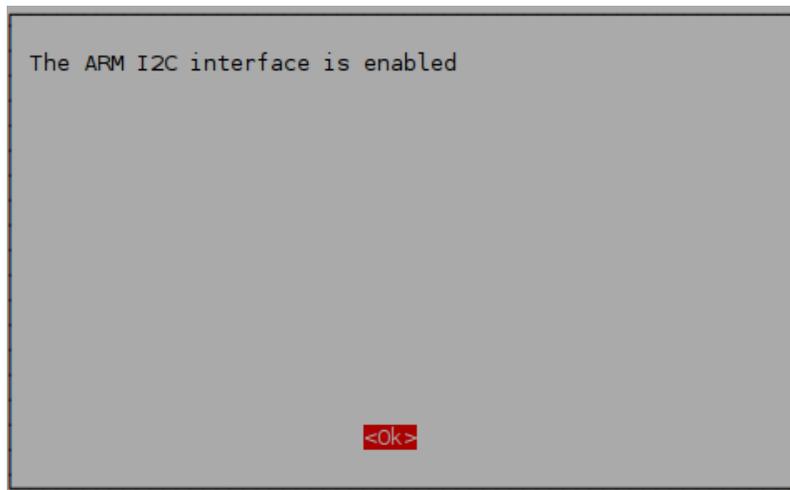
A continuación seleccionamos la opción **A7 I2C**



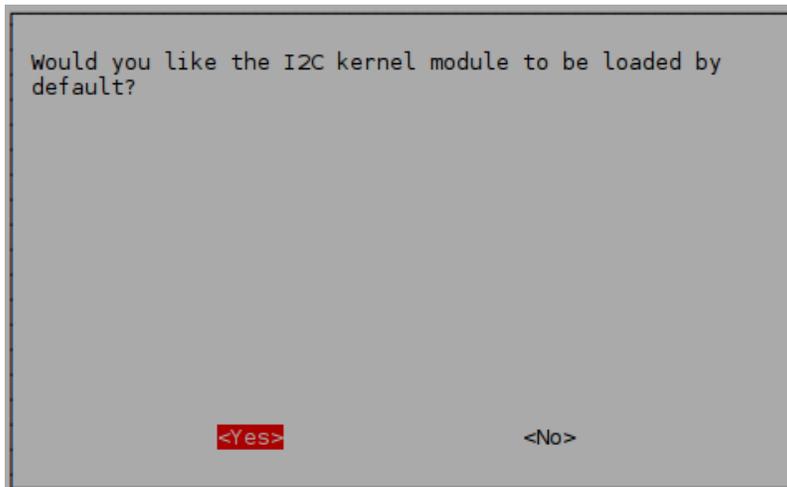
Cuando pregunta si deseamos habilitar la interfaz i2c, seleccionamos **OK**



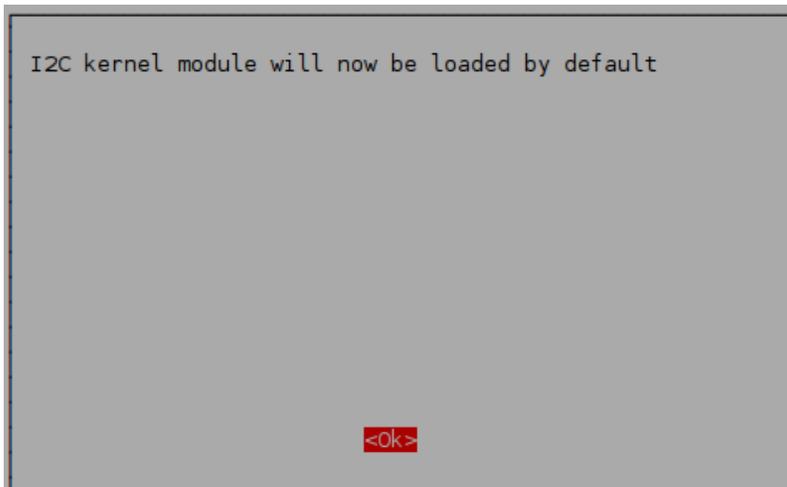
Una vez aparezca que la interfaz está habilitada seleccionamos nuevamente **OK**



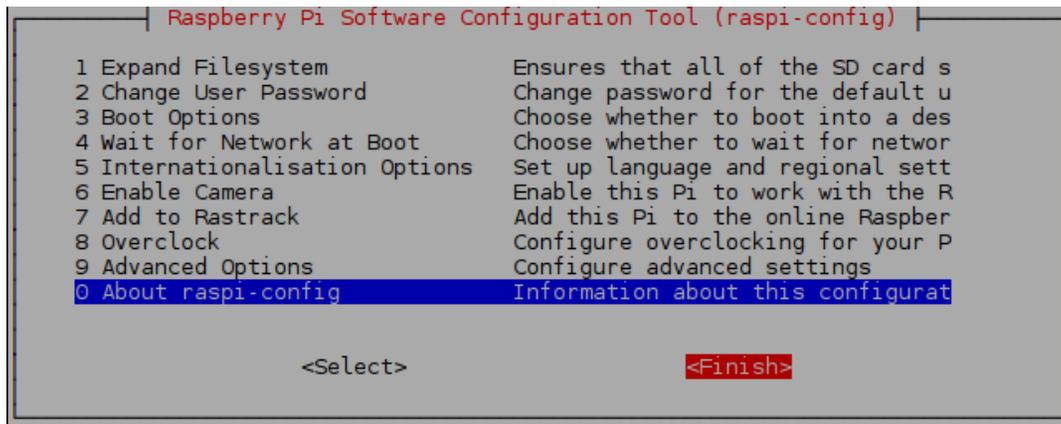
Luego pregunta si queremos que el módulo I2C sea cargado en el kernel por defecto, presionamos **YES**



Posteriormente muestra el mensaje de confirmación



Una vez se realiza este procedimiento se retorna al menú de configuración inicial y finalizamos la sesión.



Posteriormente se reinicia el sistema con el comando:

```
pi@raspberrypi:~ $ sudo reboot
```

Posteriormente se realiza la conexión del módulo DS1307 a la Raspberry Pi de acuerdo a la configuración mostrada en la Figura 93.

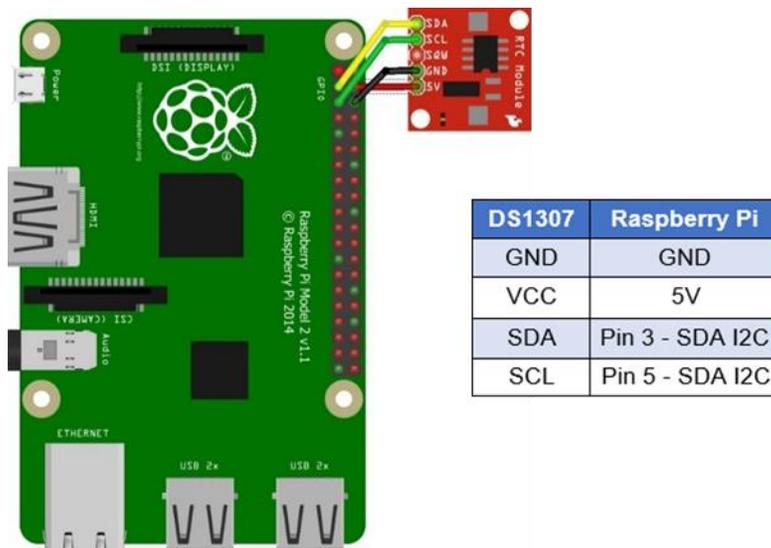


Figura 93. Conexión Raspberry Pi - Módulo RTC1307

Una vez restablecido el sistema abrimos una nueva terminal y se comprueba que la interfaz está detectando los dispositivos I2C, para lo cual se ingresa el comando:

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

Si la hora del sistema se encuentra desactualizada y no se tiene conexión a internet se puede colocar la hora manualmente utilizando el comando: **sudo date --set "2017-01-25 23:18"**

A continuación se realiza la apertura de la interfaz para probar el funcionamiento del dispositivo, por lo cual se ejecuta la siguiente combinación de comandos:

```
pi@raspberrypi:~ $ sudo modprobe rtc-ds1307
pi@raspberrypi:~ $ sudo bash
root@raspberrypi:/home/pi# echo ds1307 0x68 >/sys/class/i2c-adapter/i2c-1/new_device
root@raspberrypi:/home/pi# exit
```

Podemos verificar la hora del módulo RTC DS1307 haciendo uso del comando **sudo hwclock -r**, y compararlo con la fecha actual del sistema por medio del comando **date** como se puede observar a continuación:

```
pi@raspberrypi:~ $ sudo hwclock -r
Wed 18 Mar 1900 09:00:51 PM COT -0.918461 seconds
pi@raspberrypi:~ $ date
Wed Mar  9 19:14:57 COT 2016
```

Como se pudo observar el la hora y fecha del módulo DS1307, es diferente de la hora actual del sistema, para ajustar la hora del RTC se utiliza el comando **sudo hwclock -w**, seguido nuevamente del comando **sudo hwclock -r** para verificar la correcta escritura de la información, como se muestra a continuación:

```
pi@raspberrypi:~ $ sudo hwclock -w
pi@raspberrypi:~ $ sudo hwclock -r
Wed 09 Mar 2016 07:26:22 PM COT -0.697296 seconds
```

Con el fin de garantizar que automáticamente la fecha y hora del sistema sean actualizadas desde el módulo DS1307 editamos la configuración de los módulos y agregamos al final del archivo rtc-1307 como se ve a continuación:

```
pi@raspberrypi:~ $ sudo nano /etc/modules
```

```
GNU nano 2.2.6          Fichero: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

i2c-bcm2708
i2c-dev
rtc-ds1307
```

Posteriormente abrimos el archivo de arranque del sistema y agregamos los comandos de prueba como se muestra en las siguientes imágenes.

```
pi@raspberrypi:~ $ sudo nano /etc/rc.local
```

```
GNU nano 2.2.6          Fichero: /etc/rc.local

#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

echo Loading RTC module
sudo modprobe rtc-ds1307
sudo echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
echo Done

exit 0
```

^G Ver ayuda ^C Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^O Pos actual  
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^L PegarTxt ^T Ortografía

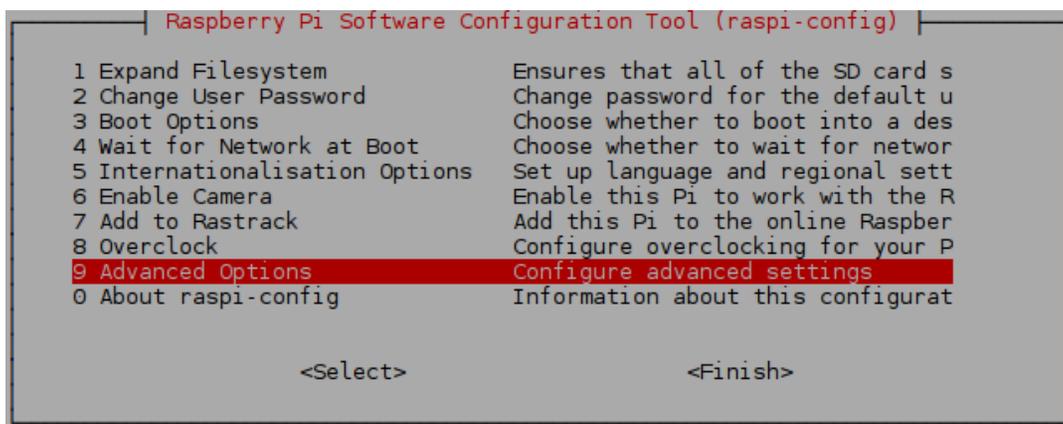
## ANEXO B. CONFIGURACIÓN INTERFAZ SPI MÓDULO ADC MCP3008

A continuación se describe el procedimiento para realizar la configuración de la interfaz SPI para poder comunicarse con el conversor analógico digital MCP3008, el cual permite adicionar a la tarjeta Raspberry Pi entradas de tipo análogo de voltaje por medio de la comunicación serial SPI.

Inicialmente se debe habilitar la interfaz SPI para lo cual se ingresa a la configuración de Raspberry con el comando:

```
pi@raspberrypi:~ $ sudo raspi-config
```

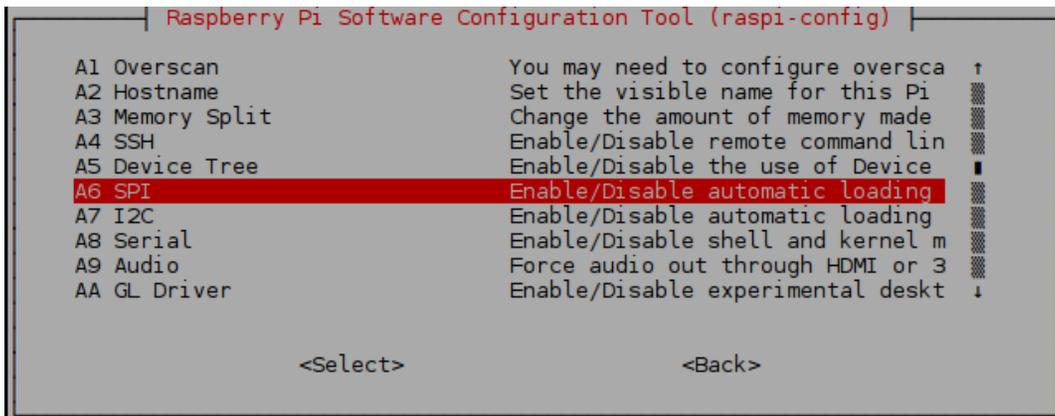
Posteriormente seleccionamos la opción **9 Advanced Options** como se muestra a continuación:



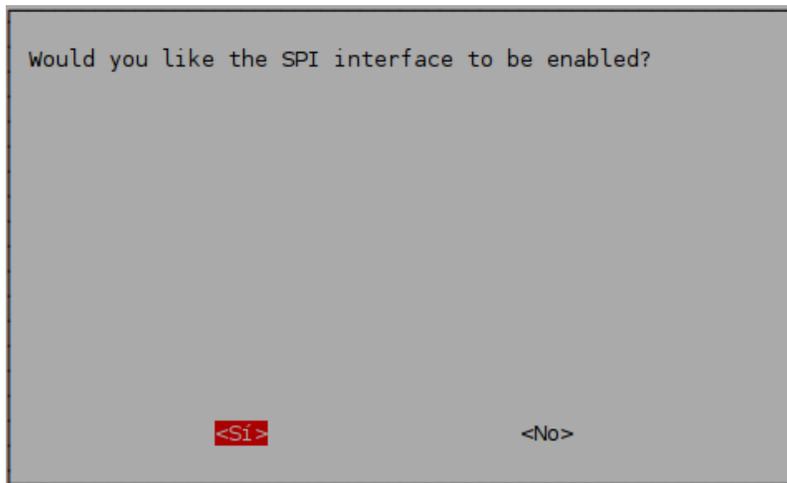
```
Raspberry Pi Software Configuration Tool (raspi-config)
1 Expand Filesystem          Ensures that all of the SD card s
2 Change User Password       Change password for the default u
3 Boot Options               Choose whether to boot into a des
4 Wait for Network at Boot   Choose whether to wait for networ
5 Internationalisation Options Set up language and regional sett
6 Enable Camera              Enable this Pi to work with the R
7 Add to Rastrack            Add this Pi to the online Raspber
8 Overclock                  Configure overclocking for your P
9 Advanced Options           Configure advanced settings
0 About raspi-config         Information about this configurat

<Select>                       <Finish>
```

A continuación seleccionamos la opción **A6 SPI**



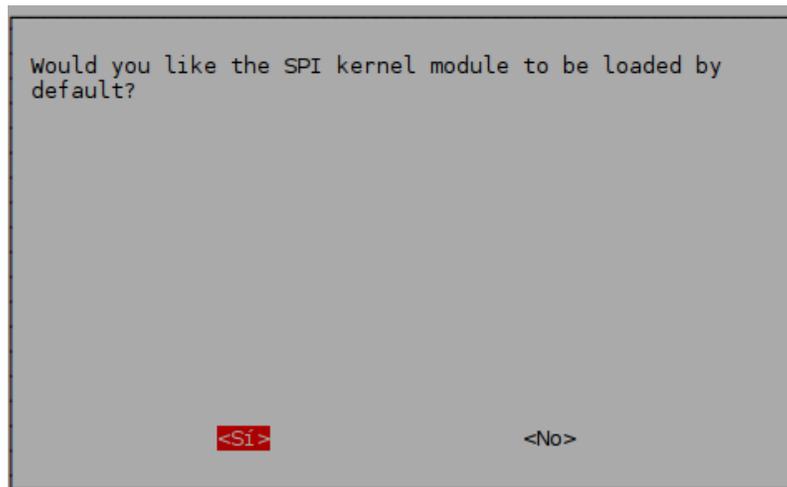
Cuando pregunta si se desea habilitar la interfaz SPI Seleccionamos **YES**



Seleccionamos la opción **ACEPTAR** cuando aparece el mensaje de confirmación que la interfaz ha sido habilitada



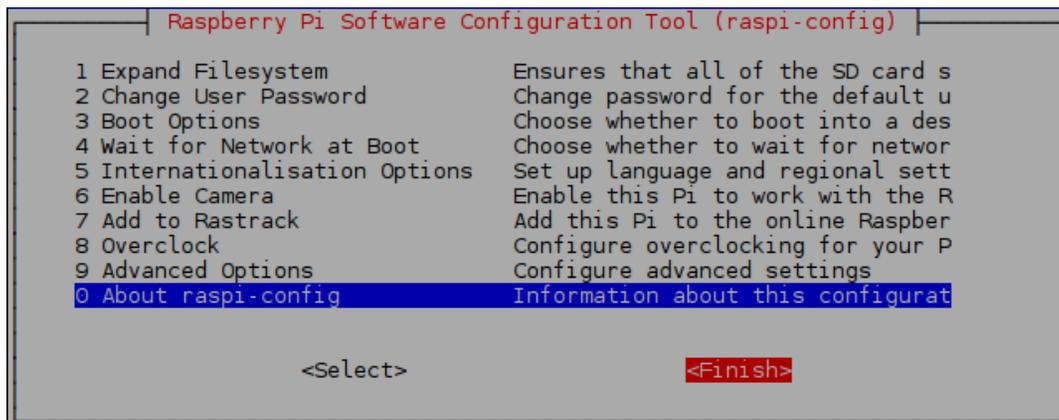
A continuación seleccionamos **YES** cuando pregunta si se desea dejar habilitada la interfaz SPI por defecto en el kernel.



Finalmente aceptamos cuando muestra la confirmación que ha sido cargada por defecto en el kernel.



Una vez realizado el procedimiento se retorna al menú de configuración inicial y finalizamos la sesión.



Posteriormente reiniciamos el sistema operativo de la Raspberry con el comando:

```
pi@raspberrypi:~ $ sudo reboot
```

A continuación en la Figura 94, se muestra el circuito de conexión implementado al cual se realizan pruebas utilizando un potenciómetro como fuente de voltaje variable.

MCP3008	Raspberry Pi
Pin 16 - VDD	5 V
Pin 15 - VREF	5V
Pin 14 - AGND	GND
Pin13 - CLK	Pin 23 - SPI_SCLK
Pin 12 - DOUT	Pin 21 - SPI_MISO
Pin 11 - DIN	Pin 19 - SPI_MOSI
Pin 10 - CS	Pin 24 - SPI_CEO
Pin 9 - DGND	GND
Pin 1 - CH0	Potenciómetro 10K

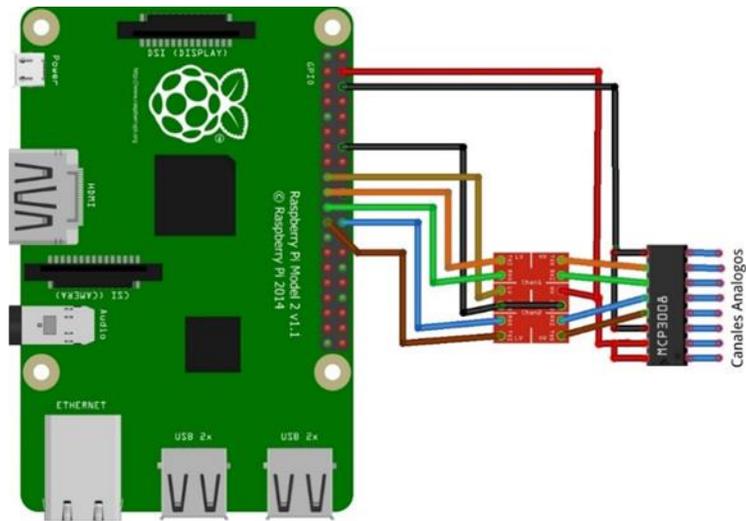


Figura 94. Conexión Raspberry Pi – Conversor MCP3008

Una vez reiniciado el sistema de la Raspberry se abre una nueva terminal y se verifica la activación del módulo SPI ejecutando el comando *Lsmod*, en el cual se debe observar que aparece el módulo “*spi\_bcm2835*” o “*spi\_bcm2708*”

```

pi@raspberrypi:~$ lsmod
Module                Size  Used by
rtc_ds1307            9402   0
cfg80211             407580  0
rfkill               16036  2 cfg80211
8192cu               519604  0
joydev                9024   0
evdev                10278   4
wl_therm              3396   0
bcm2835_gpiomem       2860   0
spi_bcm2835           7074   0
bcm2835_rng           1763   0
snd_bcm2835          19802   6
wl_gpio               3401   0
snd_pcm              73474  1 snd_bcm2835
snd_timer            18848  1 snd_pcm
wire                 24703  2 wl_gpio,wl_therm
cn                   4258   1 wire
snd                  50779  15 snd_bcm2835,snd_timer,snd_pcm
uio_pdrv_genirq       2944   0
uio                  7753   1 uio_pdrv_genirq
i2c_dev              5671   0
i2c_bcm2708          4920   0
fuse                 80694   3

```

A continuación procedemos a instalar los complementos necesario para utilizar la interfaz SPI en Python 2 y 3 para lo cual se descarga la librería por medio del siguiente comando:

```
pi@raspberrypi:~ $ wget https://github.com/Gadgetoid/py-spidev/archive/master.zip
--2016-03-22 10:54:05-- https://github.com/Gadgetoid/py-spidev/archive/master.zip
Resolviendo github.com (github.com)... 192.30.252.129
Conectando con github.com (github.com)[192.30.252.129]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización: https://codeload.github.com/Gadgetoid/py-spidev/zip/master [siguiendo]
--2016-03-22 10:54:06-- https://codeload.github.com/Gadgetoid/py-spidev/zip/master
Resolviendo codeload.github.com (codeload.github.com)... 192.30.252.160
Conectando con codeload.github.com (codeload.github.com)[192.30.252.160]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [application/zip]
Grabando a: "master.zip"

master.zip [ <=> ] 20,55K --.-KB/s en 0,1s
2016-03-22 10:54:07 (189 KB/s) - "master.zip" guardado [21039]
```

Una vez descargado el archivo se descomprime utilizando el comando:

```
pi@raspberrypi:~ $ unzip master.zip
Archive:  master.zip
26a7cb5645e2bb8f9e2f5010e23148faf70e4f69
  creating:  py-spidev-master/
  inflating:  py-spidev-master/.gitignore
  inflating:  py-spidev-master/CHANGELOG.md
  inflating:  py-spidev-master/LICENSE
  inflating:  py-spidev-master/LICENSE.md
  inflating:  py-spidev-master/MANIFEST.in
  inflating:  py-spidev-master/Makefile
  inflating:  py-spidev-master/README.md
  inflating:  py-spidev-master/setup.py
  inflating:  py-spidev-master/spidev_module.c
```

Si se desea eliminar el archivo descargado se utiliza el comando **rm master.zip**, posteriormente se abre la carpeta de ubicación del archivo:

```
pi@raspberrypi:~ $ cd py-spidev-master
```

Posteriormente se ejecutan los comandos para la instalación de la librería tanto en Python2 como Python3.

```
pi@raspberrypi:~/py-spidev-master $ sudo python setup.py install
pi@raspberrypi:~/py-spidev-master $ sudo python3 setup.py install
```

Una vez ejecutados los comandos anteriores se implementa el siguiente código (TestSPI.py) para python3 para verificar su funcionamiento.

#### Código 4. Prueba de entradas análogas con módulo MCP3008 en Python3

```
#!/usr/bin/python
import spidev
import time
import os

# SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

# Function to read SPI data from MCP3008 chip
# Channel must be an integer 0-7
def ReadChannel(channel):
    adc = spi.xfer2([1, (8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

# Function to convert data to voltage level,
# rounded to specified number of decimal places.
def ConvertVolts(data,places):
    volts = (data * 5.0) / float(1023)
    volts = round(volts,places)
    return volts

# Define sensor channels
pot = 0
# Define delay between readings
delay = 3

while True:
    # Read the light sensor data
    adc = ReadChannel(pot)
    pot_volts = ConvertVolts(adc,2)

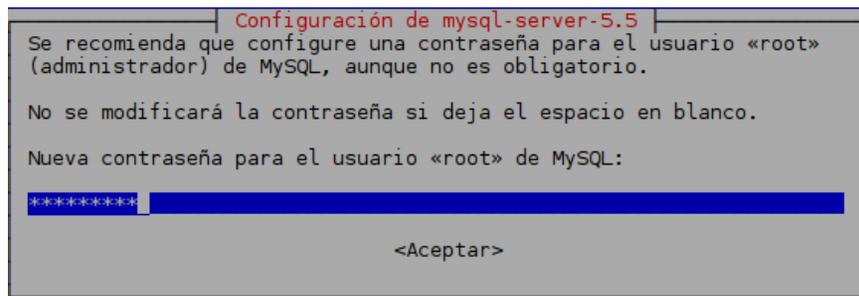
    # Print out results
    print ("-----")
    print ("Lectura ADC: ", adc)
    print("Voltaje: {}V".format(pot_volts))
    time.sleep(delay)
```

## ANEXO C. INSTALACIÓN, CONFIGURACIÓN Y MANEJO INICIAL DE BASE DE DATOS MYSQL

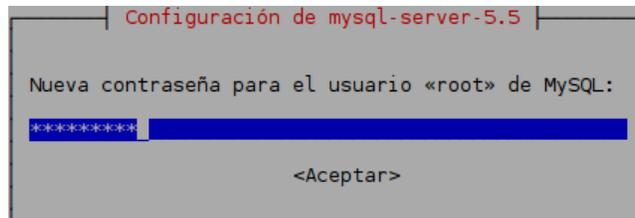
Inicialmente se realiza la instalación del servidor de base de datos y los complementos para poder ser utilizado en Python, para lo cual se utiliza el comando:

```
pi@raspberrypi:/ $ sudo apt-get install mysql-server python-mysqldb
```

Seleccionamos una contraseña, la cual servirá de acceso y será la que tenga privilegios del usuario <root> de la base de datos.



A continuación se realiza la confirmación de la contraseña de root.



Una vez finalizada la instalación del motor de MySQL, se ingresa a la aplicación a través de un terminal de consola utilizando el siguiente comando:

```
pi@raspberrypi:/ $ mysql -u root -p
Enter password: █
```

Después de ingresar la contraseña establecida en la instalación, se abre la aplicación de consola como se muestra en la siguiente imagen:

```

pi@raspberrypi:/$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.5.44-0+deb8u1 (Raspbian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Se puede verificar las bases de datos creadas por defecto con el comando:

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

```

A continuación se muestra el comando para crear una nueva base de datos, que para el ejemplo es llamada **estación**:

```

mysql> CREATE DATABASE estacion;
Query OK, 1 row affected (0.00 sec)

```

Se puede verificar la creación de la base de datos utilizando nuevamente el comando “**show databases;**”

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| estacion |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

```

Para poder realizar cualquier operación sobre la base de datos, es necesario indicarle al programa que se desea utilizar dicha base de datos, para lo cual se utiliza el comando **USE** como se muestra a continuación.

```
mysql>
mysql> USE estacion
Database changed
```

Dentro de la base de datos se puede verificar las tablas creadas dentro de esta, haciendo uso del comando **show tables**; a continuación se observa que la base de datos se encuentra vacía:

```
mysql>
mysql> show tables;
Empty set (0.00 sec)
```

Para crear una tabla dentro de la base de datos, es necesario conocer los tipos de datos que se desean almacenar, asignarles un nombre, tamaño y llave primaria, a continuación se muestra un ejemplo de creación de la tabla valores, utilizada en el desarrollo del proyecto:

```
mysql> CREATE TABLE valores (id INT NOT NULL AUTO_INCREMENT, fecha VARCHAR(10),
hora VARCHAR(10), id_sensor INT, dato VARCHAR(10), PRIMARY KEY(id));
Query OK, 0 rows affected (1.33 sec)
```

Se puede verificar la creación de la tabla utilizando nuevamente el comando **show tables**;

```
mysql> show tables;
+-----+
| Tables_in_estacion |
+-----+
| valores             |
+-----+
1 row in set (0.00 sec)
```

Es posible también verificar la configuración de la tabla creada utilizando el comando **DESCRIBE**, seguido del nombre de la tabla como se muestra a continuación:

```
mysql>
mysql> DESCRIBE valores;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRI | NULL | auto_increment |
| fecha | varchar(10) | YES | | NULL | |
| hora  | varchar(10) | YES | | NULL | |
| id_sensor | int(11) | YES | | NULL | |
| dato  | varchar(10) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Con el fin de realizar exportaciones de la información almacenada en la base de datos es necesario realizar modificaciones en los permisos de escritura de la configuración de MySQL, para lo cual se ingresa al archivo de permisos utilizando el comando:

```
pi@pi-VirtualBox: ~  
pi@pi-VirtualBox:~$ sudo nano /etc/apparmor.d/usr.sbin.mysqld
```

Se edita este archivo adicionando las rutas deseadas para realizar exportaciones y el permiso de escritura como se muestra en la siguiente imagen:

```
GNU nano 2.2.6 File: /etc/apparmor.d/usr.sbin.mysqld  
  
/var/log/mysql.err rw,  
/var/lib/mysql/ r,  
/var/lib/mysql/** rwk,  
/var/lib/mysql-files/ r,  
/var/lib/mysql-files/** rwk,  
/var/log/mysql/ r,  
/var/log/mysql/* rw,  
/var/run/mysqld/mysqld.pid rw,  
/var/run/mysqld/mysqld.sock w,  
/run/mysqld/mysqld.pid rw,  
/run/mysqld/mysqld.sock w,  
/home/pi/Documents/* w,  
/home/pi/Desktop/* w,  
  
/sys/devices/system/cpu/ r,  
  
# Site-specific additions and overrides. See local/README for details.  
#include <local/usr.sbin.mysqld>  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Una vez realizado el cambio en el archivo se guardan los cambios y se reinicia el servicio de la base de datos utilizando el comando:

```
pi@pi-VirtualBox: ~  
pi@pi-VirtualBox:~$ sudo /etc/init.d/apparmor reload
```

Posteriormente es necesario otorgar permisos de lectura y/o escritura a las ubicaciones donde se desean exportar los archivos desde la base de datos, en este caso se desea exportar archivos a la carpeta **/home/pi/Desktop/**, para la cual se utiliza el siguiente comando:

```
pi@raspberrypi:~ $ sudo chmod 1777 /home/pi/Desktop/  
pi@raspberrypi:~ $
```

Adicionalmente es necesario cambiar el estado del archivo privado de seguridad dentro de la configuración de MySQL para lo cual se edita el archivo **/etc/mysql/my.cnf** con el comando:

```
pi@pi-VirtualBox: ~  
pi@pi-VirtualBox:~$ sudo nano /etc/mysql/my.cnf
```

Al final del archivo es necesario añadir la instrucción **secure\_file\_priv=""** como se muestra a continuación:

```
pi@pi-VirtualBox: ~  
GNU nano 2.2.6 File: /etc/mysql/my.cnf Modified  
key_buffer = 16M  
##  
# * IMPORTANT: Additional settings that can override those from this file!  
# The files must end with '.cnf', otherwise they'll be ignored.  
#  
!includedir /etc/mysql/conf.d/  
[mysqld]  
secure_file_priv=""
```

Finalmente se guardan los cambios y se reinicia el sistema operativo utilizando el comando:

```
pi@raspberrypi:~ $ sudo reboot
```

Para verificar el funcionamiento de la exportación de datos se realiza una exportación de prueba, preferiblemente de tablas que tengan almacenados datos, utilizando el comando:

```
mysql> SELECT * FROM valores INTO OUTFILE "/home/pi/Desktop/archivo3.csv" FIELDS  
TERMINATED BY ";" LINES TERMINATED BY "\n";  
Query OK, 2178 rows affected (0.03 sec)
```

## ANEXO D. ESTUDIO DEL DIMENSIONAMIENTO DE UN SISTEMA DE ALIMENTACIÓN SOLAR FOTOVOLTAICO PARA EL PROTOTIPO DESARROLLADO.

Con el fin de utilizar el prototipo desarrollado en locaciones remotas donde no se cuente con un sistema de alimentación provisto de la red eléctrica convencional, se plantea el uso de energía solar fotovoltaica para la alimentación de los componentes, los sensores y el módem utilizados.

Basados en el manual para instalaciones fotovoltaicas autónomas [56], se realizan los cálculos correspondientes para determinar el número de paneles solares necesarios, la capacidad de la batería y regulador de carga.

Se plantea un diseño teniendo en cuenta el consumo de los dispositivos utilizados en la implementación del prototipo, por lo cual se consultaron las fichas técnicas correspondientes y se realizaron mediciones de corriente para obtener la Tabla 20, donde se relaciona el consumo de cada uno de los elementos, donde **además se añade un 20% adicional**, utilizado como margen de seguridad recomendado para la protección de los elementos.

Elemento	Consumo (A)	Potencia Unitaria (W)	Horas	Consumo (Wh/día)	Consumo (Wh/día) + 20% de seguridad
Raspberry Pi	0,160	1,920	14	26,880	32,256
Display 7"	0,250	3,000	14	42,000	50,400
Prototipo Tarjeta	0,020	0,240	14	3,360	4,032
Módem GPRS	0,250	3,000	14	42,000	50,400
Sensor Dirección y Velocidad del Viento	0,167	2,000	14	28,000	33,600
Sensor Temperatura y Humedad	0,008	0,096	14	1,344	1,613
Sensor Presión Atmosférica	0,004	0,048	14	0,672	0,806
Total:	0,859	10,304		144,3	173,1

Tabla 20. Consumos de los elementos utilizados en el prototipo.

Para realizar el cálculo del consumo medio diario, se debe tener en cuenta las posibles pérdidas en los elementos utilizados en el sistema de alimentación, por lo cual se

considera el **rendimiento de la batería** en un 95% y el **rendimiento de los conductores** a un 100%, con lo cual se obtienen los resultados de la Ecuación 1, donde es importante señalar que el prototipo utiliza únicamente cargas de corriente continua, por lo cual no se tiene en cuenta para los cálculos las cargas AC ni los parámetros para un inversor, asumiéndolos como 0 Wh/día.

$$L_{md} = \frac{L_{md.DC} + \frac{L_{md.AC}}{\eta_{Inv}}}{\eta_{bat} * \eta_{con}} = \frac{173.1 + 0}{0.95 * 1} = 182.2 \text{ Wh/día} \quad (1)$$

Donde:

$L_{md}$  = Consumo Medio diario

$L_{md.DC}$  = Consumo de Corriente DC

$L_{md.AC}$  = Consumo de Corriente AC = 0

$\eta_{Inv}$  = Rendimiento del Inversor AC

$\eta_{Bat}$  = Rendimiento de la Batería (95%)

$\eta_{con}$  = Rendimiento de los Conductores (100%)

Una vez calculado el consumo medio diario, que es de 182.2 Wh/día, se procede a realizar el cálculo del número de paneles que necesitaría el sistema para funcionar. Para esto se debe conocer el tipo de panel solar a utilizar, que en este caso se realiza con un panel solar **Yingli Solar YL80P**, el cual cuenta con las características mostradas en la Figura 95, donde se destacan la potencia pico de 80W y la corriente en corto circuito de 5.04 A.



Figura 95. Panel solar YingliSolar YL80P y características

De igual forma es importante conocer las condiciones de radiación del sitio donde se instala el sistema, para este caso el diseño se realiza para las condiciones ambientales de la ciudad de Bogotá, por lo cual basado en la información obtenida del atlas de radiación nacional [57], relacionada en la Tabla 21, se decidió utilizar el promedio acumulado diario de radiación para la ciudad durante el mes de Abril debido a que es el mes con menor promedio durante el año con un valor de 3270,4 Wh/m<sup>2</sup>.

El cálculo del número de paneles se realiza de acuerdo a la formula presentada en la ecuación 2, donde se obtiene que para este caso es suficiente utilizar un (1) panel con las características expuestas anteriormente.

$$N_T = \frac{L_{md}}{P_{MPP} * HPS_{crit} * PR} = \frac{182.2}{80 * 3.2704 * 0.90} = 0.774 \approx \mathbf{1 \text{ Panel}} \quad (2)$$

Donde:

$N_T = N$  umero total de Paneles

$L_{md} =$  Consumo Medio diario

$P_{MPP} =$  Potencia Pico Panel Solar

$HPS_{crit} =$  Horas Pico Sol durante el mes critico en la ubicacion seleccionada

$PR =$  Factor Global de Fucnionamiento (0.90 por defecto)

A continuación se realiza el cálculo de la capacidad nominal de la batería, necesaria para el funcionamiento adecuado del sistema. Para esto se debe tener en cuenta la profundidad de descarga máxima permitida antes de la desconexión de las cargas conectadas. Generalmente se consideran dos tipos de descargas, la **Profundidad de descarga máxima diaria**, la cual es considerada en un ciclo de funcionamiento diario (generalmente entre el 15 y 20%), y la **Profundidad de descarga máxima de ciclo estacionario** (generalmente 70%), donde además se tiene en cuenta el periodo de tiempo (de 4 a 10 días) en que la batería puede suministrar alimentación sin recibir carga del panel solar.

En la ecuación 3, se realiza el cálculo de la capacidad nominal en función de la descarga máxima diaria, donde se tiene en cuenta una profundidad del 15% y se asume un factor de corrección por temperatura igual a 1.

$$C_{nd}(Wh) = \frac{L_{md}}{P_{D \max d} * F_{CT}} = \frac{182.2}{0.15 * 1} = 1214.7 Wh \quad (3)$$

Donde:

$C_{nd}(Wh)$  = Capacidad Nominal (descarga máxima diaria)

$L_{md}$  = Consumo Medio diario

$P_{D \max d}$  = Profundidad de descarga máxima diaria (15%)

$F_{CT}$  = Factor de corrección por temperatura (1 por defecto)

En la ecuación 4 se expresa la capacidad nominal en amperios hora (Ah), en función de la descarga máxima diaria, teniendo en cuenta el voltaje de la batería.

$$C_{nd}(Ah) = \frac{C_{nd}(Wh)}{V_{Bat}} = \frac{1214.7}{12} = 101,2 Ah \quad (4)$$

En la ecuación 5, se realiza el cálculo de la capacidad nominal en función de la descarga máxima de ciclo estacionario, donde se tiene en cuenta 4 días de autonomía para el sistema, una profundidad del 70% y suponiendo un factor de corrección por temperatura igual a 1.

$$C_{ne}(Wh) = \frac{L_{md} * N_{días}}{P_{D \max e} * F_{CT}} = \frac{182.2 * 5}{0.7 * 1} = 1301,5 Wh \quad (5)$$

Donde:

$C_{ne}(Wh)$  = Capacidad Nominal (descarga máxima estacional)

$L_{md}$  = Consumo Medio diario

$P_{D \max e}$  = Profundidad de descarga máxima estacional (70%)

$N_{días}$  = Numero de días de Autonomía

$F_{CT}$  = Factor de corrección por temperatura (1 por defecto)

En la ecuación 6 se expresa la capacidad nominal en amperios hora (Ah) en función de la descarga máxima estacional teniendo en cuenta el voltaje de la batería.

$$C_{ne}(Ah) = \frac{C_{ne}(Wh)}{V_{Bat}} = \frac{1301,5}{12} = 108,5 Ah \quad (6)$$

El criterio de selección de la capacidad nominal de la batería se realiza escogiendo el valor mayor entre los cálculos realizados, teniendo en cuenta las profundidades máximas (diaria y estacional) ya que la capacidad en función de la descarga estacional obtenida fue

de 108,5 Ah, se decide utilizar una batería de uso comercial con una capacidad de por lo menos **110 Ah**.

Finalmente se realiza el cálculo para determinar la capacidad del regulador basado en la corriente de entrada y la corriente de salida que debe soportar. En la ecuación 7 se determina la corriente de entrada que debe soportar el regulador teniendo en cuenta un factor de seguridad (1.25) para evitar daños en el regulador, la corriente de corto circuito proporcionada por las características técnicas del panel solar y el número total de paneles a utilizar.

$$I_{Entrada} = 1.25 * I_{Panel.SC} * N_T = 1.25 * 5.04 * 1 = 6.3A \quad (7)$$

Donde:

$I_{Entrada}$  = Corriente de Entrada del Regulador Solar

$I_{Panel.SC}$  = Corriente de Corto Circuito del Panel Solar

$N_T$  = Número total de Paneles

Para el cálculo de la corriente de salida se debe tener en cuenta la potencia en *Watts* de las cargas, obtenidos en la tabla de consumo y el voltaje de la batería. En la ecuación 8 se muestra el cálculo de la corriente de salida, donde nuevamente al no tener cargas de corriente alterna, esta parte de la ecuación se asume como 0W.

$$I_{Salida} = 1.25 \frac{P_{DC} + \frac{P_{AC}}{\eta_{Inv}}}{V_{Bat}} = 1.25 \frac{10,304 + 0}{12} = 1.1 A \quad (8)$$

Donde:

$I_{Salida}$  = Corriente de Salida del Regulador Solar

$P_{DC}$  = Potencia en Watts de las cargas DC

$P_{AC}$  = Potencia en Watts de las cargas AC

$\eta_{Inv}$  = Rendimiento del Inversor AC

$V_{Bat}$  = Voltaje de la Bateria

Por lo tanto el regulador de carga debe soportar una corriente de **6.3A** a su entrada y una corriente de **1.1A** a su salida.

Tabla 21. Promedio horario de la radiación para la estación universidad nacional de Bogotá.

ESTACION UNIVERSIDAD NACIONAL DE COLOMBIA (BOGOTÁ)												
PROMEDIO HORARIO DE LA RADIACION (Wh/m <sup>2</sup> )												
HORA	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
0-1	0,2	0,3	0,3	0,2	0,2	0,2	0,3	0,4	0,5	0,2	0,3	0,3
1-2	0,3	0,2	0,3	0,3	0,2	0,2	0,4	0,5	0,4	0,2	0,4	0,4
2-3	0,2	0,2	0,2	0,2	0,2	0,2	0,5	0,5	0,4	0,2	0,3	0,3
3-4	0,2	0,2	0,1	0,3	0,2	0,3	0,3	0,4	0,5	0,2	0,3	0,3
4-5	0,1	0,1	0,3	0,1	0,2	0,3	0,3	0,4	0,5	0,2	0,4	0,3
5-6	0,2	0,1	0,3	1,3	2,5	1,9	1,2	1,2	2,0	3,0	2,7	0,7
6-7	31,8	28,1	39,5	52,9	66,3	62,0	55,2	50,6	60,2	71,0	61,4	50,2
7-8	184,9	162,3	149,3	177,2	188,3	188,0	191,0	172,9	180,6	205,6	198,3	208,9
8-9	351,3	342,2	264,3	309,3	294,4	291,3	313,7	290,8	284,5	346,4	356,3	378,8
9-10	507,6	455,8	366,7	388,1	391,4	389,5	399,7	387,8	399,5	457,6	458,6	504,3
10-11	589,9	543,7	469,8	425,6	431,5	447,8	476,8	467,0	460,2	501,8	519,1	545,5
11-12	607,4	544,1	483,3	433,2	455,7	497,2	520,6	491,4	489,4	458,0	488,2	533,5
12-13	560,1	503,8	433,2	412,9	464,7	490,0	544,6	494,3	481,1	442,4	414,1	491,1
13-14	475,7	465,6	389,8	365,9	404,2	476,3	499,4	457,5	432,5	393,3	355,3	423,4
14-15	375,9	357,0	313,5	307,7	334,2	391,8	404,2	392,0	398,2	315,9	274,7	348,9
15-16	304,4	275,2	231,1	225,1	245,4	275,5	292,9	305,2	298,4	209,9	188,3	254,7
16-17	171,2	145,7	145,0	135,4	143,1	158,6	177,2	168,1	168,4	111,0	90,2	124,9
17-18	46,4	48,7	47,7	35,6	35,7	46,3	59,4	48,9	34,5	16,7	13,0	26,0
18-19	0,5	0,9	0,7	0,2	0,3	0,5	1,1	0,9	0,4	0,3	0,4	0,4
19-20	0,3	0,2	0,2	0,2	0,2	0,1	0,2	0,3	0,2	0,3	0,3	0,2
20-21	0,3	0,3	0,2	0,2	0,2	0,1	0,2	0,2	0,2	0,1	0,3	0,3
21-22	0,2	0,2	0,2	0,2	0,2	0,1	0,2	0,3	0,4	0,2	0,3	0,2
22-23	0,2	0,2	0,1	0,2	0,2	0,2	0,2	0,4	0,3	0,2	0,2	0,3
23-0	0,1	0,2	0,2	0,2	0,2	0,1	0,3	0,3	0,5	0,2	0,3	0,2
Acumulada diaria	4207,3	3873,1	3334,2	3270,4	3457,9	3716,7	3937,1	3728,5	3690,0	3533,0	3420,7	3891,2



## ANEXO E. CODIGOS UTILIZADOS PARA EL DESARROLLO DE LA INTERFAZ GRÁFICA DE CONFIGURACION Y VISUALIZACION

Para el desarrollo de la aplicación gráfica se utilizó la plataforma de software **MONO** la cual cuenta con un compilador de **C#**, utilizado para el desarrollo de la aplicación, junto con la librería **GTK#**, la cual es la herramienta de mono para entornos de aplicaciones GUI. Para desarrollar la interfaz gráfica de configuración y visualización se utiliza el entorno de desarrollo integrado (IDE) **MonoDevelop**, el cual trabaja con mono y permite la compilación y ejecución de código en C#, en el cual se crean los códigos para el desarrollo de la aplicación gráfica, en el cual se instalan componentes necesarios para el adecuado funcionamiento, como por ejemplo los paquetes **“OxyPlot.Core”** y **“Oxyplot.GtkSharp”** para la creación de gráficas lineales, el paquete **”netfx-System.Xml.XmlSerializer”** para la lectura y escritura de archivos en formato XML y el conector **“MySql.Data”** para la interacción con bases de datos desarrolladas en MySql.

El Programa principal se compone de 3 códigos fuente descritos a continuación:

- **MainWindow.cs**: este código está compuesto por 1380 líneas, encargadas de crear los objetos correspondientes para la visualización de pestañas, y botones, además se encarga de la interacción con la base de datos realizando la lectura y escritura de la información almacenada.  
**Configuration.cs**: este código está compuesto por 289 líneas en las cuales se declaran e inicializan las variables utilizadas para la creación del archivo de configuración XML.
- **Common.cs**: este código está compuesto por 80 líneas, en el cual se encuentran los métodos encargados para cargar y guardar la información en el archivo de configuración, igualmente se encarga de gestionar los objetos para la visualización de la consola de ayuda.

Adicionalmente el compilador del programa genera automáticamente, diferentes códigos internos junto con los cuales se completa las herramientas necesarias para el adecuado funcionamiento de la interfaz gráfica.

## ANEXO F. CODIGOS UTILIZADOS PARA EL FUNCIONAMIENTO DEL PROTOTIPO

### F.1 PLANTILLA XML HYDRAS3

A continuación se presenta la plantilla en formato XML utilizada para hacer la integración de los datos enviados por el prototipo desarrollado al software de visualización Hydras3. Esta plantilla contiene la información asociada a la identificación de la estación como el código y nombre de la estación, y la información de los sensores a recibir como el código del sensor, nombre y unidad de la variable.

#### Código 5. Plantilla XML utilizada para la integración de datos a Hydras3

```
<STATION>
<DEVICETYPE>RASPBERRY PI 2</DEVICETYPE>
<STATIONID>0088112901</STATIONID>
<STATIONNAME>ECI JULIO GARAVITO</STATIONNAME>
<STATIONUTC>-05:00</STATIONUTC>
<SENSOR>
<SENSORID>0068</SENSORID>
<SENSORNAME>Temperatura del Aire</SENSORNAME>
<UNIT>°C</UNIT>
<RIGHTDIGITS>1</RIGHTDIGITS>
<AXISLABEL>Temperatura del Aire</AXISLABEL>
<INTERVAL>0</INTERVAL>
</SENSOR>
<SENSOR>
<SENSORID>0027</SENSORID>
<SENSORNAME>Humedad del Aire</SENSORNAME>
<UNIT>%</UNIT>
<RIGHTDIGITS>0</RIGHTDIGITS>
<AXISLABEL>Humedad del Aire</AXISLABEL>
<INTERVAL>0</INTERVAL>
</SENSOR>
<SENSOR>
<SENSORID>0255</SENSORID>
<SENSORNAME>Presion Atmosferica</SENSORNAME>
<UNIT>HPa</UNIT>
<RIGHTDIGITS>1</RIGHTDIGITS>
<AXISLABEL>Presion Atmosferica</AXISLABEL>
<INTERVAL>0</INTERVAL>
</SENSOR>
<SENSOR>
<SENSORID>0240</SENSORID>
<SENSORNAME>Precipitacion Acumulada 10min</SENSORNAME>
<UNIT>mm</UNIT>
<RIGHTDIGITS>1</RIGHTDIGITS>
<AXISLABEL>Precipitacion Acumulada 10min</AXISLABEL>
```

```

    <INTERVAL>0</INTERVAL>
</SENSOR>
<SENSOR>
  <SENSORID>0103</SENSORID>
  <SENSORNAME>Velocidad del Viento</SENSORNAME>
  <UNIT>m/s</UNIT>
  <RIGHTDIGITS>1</RIGHTDIGITS>
  <AXISLABEL>Velocidad del Viento</AXISLABEL>
  <INTERVAL>0</INTERVAL>
</SENSOR>
<SENSOR>
  <SENSORID>0104</SENSORID>
  <SENSORNAME>Direccion del Viento</SENSORNAME>
  <UNIT>°N</UNIT>
  <RIGHTDIGITS>0</RIGHTDIGITS>
  <AXISLABEL>Direccion del Viento</AXISLABEL>
  <INTERVAL>0</INTERVAL>
</SENSOR>
</STATION>

```

## F.2 PROGRAMA DE APLICACIÓN: APLICACIONDCP\_01.PY

A continuación se presenta el código desarrollado en Python3, de la aplicación principal del prototipo, el cual es el encargado de realizar la lectura de los sensores análogos, digitales y SDI12, de igual forma realiza la conexión y almacenamiento de la información en la base de datos y de acuerdo a las condiciones periódicas de tiempo envía la información al servidor remoto.

### Código 6. Aplicación principal del prototipo desarrollado en Python3

```

from PrincipalDCP import *
from XMLRead import *

def cuentaCH8(x):
    global contadorCH8
    time.sleep(0.5)
    contadorCH8 = contadorCH8 + ResCH8
    contadorCH8=round(contadorCH8,int(DecimalesCH8))
GPIO.add_event_detect(pinCH8, GPIO.FALLING, callback=cuentaCH8)

def cuentaCH9(x):
    global contadorCH9
    time.sleep(0.5)
    contadorCH9 = contadorCH9 + ResCH9
    contadorCH9=round(contadorCH9,int(DecimalesCH9))
GPIO.add_event_detect(pinCH9, GPIO.FALLING, callback=cuentaCH9)

if ser.isOpen():
    ser.close()

```

```

        print ('Serial Close')

ser.open()
print ('serial Open')
time.sleep(0.6)

while True:
    actual = datetime.now() # Lectura fecha-hora Actual
    fecha = actual.strftime(formato1) #fecha formato plano
    hora = actual.strftime(formato2) #Hora formato plano

    #Lectura de datos en Pantalla
    if (actual.second%Muestreo)==0: #Condicion Para Visualizacion
Periodica
        datos=[]
        id_sensor=[]
        for i in range(len(Analogos)):
            datos.append(analogmap(Analogos[i][0], Analogos[i][5],
Analogos[i][6], Analogos[i][7], Analogos[i][8], int(Analogos[i][4])))
            id_sensor.append(Analogos[i][2])
        for i in range(len(Digitales)):
            if Digitales[i][0]==8:
                datos.append(contadorCH8)
            elif Digitales[i][0]==9:
                datos.append(contadorCH9)
            id_sensor.append(Digitales[i][2])
        for i in range(len(Seriales)):
            datos.append(SDI12(int(Seriales[i][5]), int(Seriales[i][6]),
int(Seriales[i][4])))
            id_sensor.append(Seriales[i][2])

        actualizaDB(id_sensor, datos)

    #Almacenamiento y Transmision de Datos
    if (actual.minute%TiempoTx)==0 and actual.second==0: #Condicion Para
Transmision Periodica
        print('Ingresa a condicion periodica de Tx')

        #Almacenamiento en Base de Datos
        if(SaveDB=='true'):
            print('SAVE DB')
            insertarDB(CodigoEstacion, id_sensor, datos, fecha, hora)

        # Creacion archivo .MIS
        if(ActivaTx=='true'):
            Archivo=crearMIS(CodigoEstacion, id_sensor, datos, fecha,
hora)

            # Enviar Archivo FTP
            enviarFTP(Archivo)

        contadorCH8=0.0
        contadorCH9=0.0

time.sleep(1)

```

### F.3 LIBRERÍA DE FUNCIONES: PRINCIPALDCP.PY

A continuación se presenta la librería de funciones creada, con el fin de agrupar diferentes funciones que son usadas en el programa de aplicación principal. Esta librería se encarga de realizar las configuraciones iniciales para el uso de los pines GPIO, y la comunicación SPI y serial, así mismo contiene funciones encargadas de la conexión a la base de datos, la parametrización de variables análogas, la creación del archivo .MIS, la conexión al servidor FTP y la captura de imágenes.

#### Código 7. Librería de funciones creadas para el prototipo: "PrincipalDCP.py"

```
#Libreria Princial DCP

# Llamado a Librerias
import sys
import time
from datetime import datetime
import spidev
import RPi.GPIO as GPIO
import os
import os.path
import cymysql
import random
import ftplib
import serial
from XMLRead import *
from picamera import PiCamera
from subprocess import call

#Definicion de Variables
formato1 = "%Y%m%d"
formato2 = "%H%M%S"
formato3 = "%Y/%m/%d"
formato4 = "%H:%M:%S"

#Configuracion Puerto Serial
ser = serial.Serial(
    port='/dev/ttyUSB0',
    baudrate = 9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)

### Configuracion SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

#Configuracion GPIO
```

```

GPIO.setmode(GPIO.BCM)

#Conexion con base de datos MySQL
db = cymysql.connect(host="localhost", user="root", passwd="raspberr",
db="station")
cursor = db.cursor()

#Definicion de Funciones

def readSPI(channel):
    adc = spi.xfer2([1, (8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    volts = (data * 5.0) / float(1023)
    time.sleep(0.0001)
    return volts

def analogmap(channel, Inmin, Inmax, Outmin, Outmax, decimal):
    volts = readSPI(channel)
    if volts>Inmax:
        volts=Inmax
    #calculo=((volts-Inmin)*(Outmax-Outmin))/(Inmax-Inmin))+Outmin
    calculo=(volts*((Outmax-Outmin)/(Inmax-Inmin)))+Outmin
    out=round(calculo,decimal)
    #time.sleep(0.0001)
    return out

def bateria(channel,decimal):
    volts = readSPI(channel)
    batery=volts/float(0.32)
    out=round(batery,decimal)
    #time.sleep(0.0001)
    return out

#Funcion lectura SDI12
def SDI12(direccion,posicion,decimal):

    ser.flushInput
    ser.flushOutput

    cadena=str(direccion)+str(posicion)
    ## print(cadena)

    ser.write(cadena.encode())

    x=b''
    c=0
    while x==b'' and c<5:
        x=ser.readline()
        #print(x)
        c+=1
    ## print("valor final de x=")
    ## print(x)
    y=x.decode(encoding='UTF-8')
    print(y)

```

```

## print(type(y))

ser.flushInput
ser.flushOutput

out=y
return out

#Configuracion Pines Entradas Digitales
pinCH8=20
GPIO.setup(pinCH8, GPIO.IN, pull_up_down=GPIO.PUD_UP)
contadorCH8=0.0

pinCH9=21
GPIO.setup(pinCH9, GPIO.IN, pull_up_down=GPIO.PUD_UP)
contadorCH9=0.0

#Almacenar en Base de Datos
def insertarDB(id_estacion, id_sensor, datos, fecha, hora):
    print('Inicia Conexion a DB')
    insertar="INSERT INTO datos (id_estacion, fecha, hora, id_sensor,
dato)VALUES(%s, %s, %s, %s, %s)"
    for i in range(len(datos)):
        cursor.execute(insertar,(id_estacion,fecha, hora, id_sensor[i],
datos[i]))
        db.commit()
    print('Almacenado Correctamente en DB')

#Actualizar en Base de Datos
def actualizaDB(id_sensor, datos):
    print('Inicia Conexion a DB Update')
    actualizar="UPDATE variables SET dato=%s WHERE id_sensor=%s"
    for i in range(len(datos)):
        cursor.execute(actualizar,(datos[i],id_sensor[i]))
        db.commit()
    print('Actualiza Correctamente en DB')

#Funcion Crear .MIS
def crearMIS(id_estacion, id_sensor, datos, fecha, hora):
    id_estacion_str='00'+str(id_estacion)

    id_sensor_str=[]
    for i in range(len(id_sensor)):
        if len(str(id_sensor[i]))==1:
            id_sensor_str.append('000'+str(id_sensor[i]))
        elif len(str(id_sensor[i]))==2:
            id_sensor_str.append('00'+str(id_sensor[i]))
        elif len(str(id_sensor[i]))==3:
            id_sensor_str.append('0'+str(id_sensor[i]))
        else:
            id_sensor_str.append(str(id_sensor[i]))

    datos_str=[]
    for i in range(len(datos)):

```

```

    datos_str.append(str(datos[i]))

    Archivo = id_estacion_str+'_'+ fecha+ hora+'.MIS' #Nombre del Archivo a
Crear

    fo = open('/home/pi/Documents/ProyectoGrado/FTP/'+ Archivo ,"w+") #Ruta
para Crear El Archivo
    print ("Creando Archivo..." )

    for i in range(len(datos)):
        fo.write('<STATION>'+ id_estacion_str+'</STATION><SENSOR>'+
id_sensor_str[i] +'</SENSOR><DATEFORMAT>YYYYMMDD</DATEFORMAT>\n')
        fo.write(fecha+ ';' + hora+ ';' + datos_str[i]+ '\n')
    fo.close() # Finaliza Creacion del Archivo
    print ("Archivo Local Creado" )
    return Archivo

#Funcion Enviar FTP
def enviarFTP(Archivo):
    try:
        print ("Intentando Abrir FTP "+Archivo)
        ftp_connection = ftplib.FTP('ftp.ideam.gov.co', 'ideamgrs',
'20120816JCFB') #Conexion con Servidor
        print ("FTP conectado OK")
        Directorio = "/sinopticas" #Nombre de Cartpeta
        ftp_connection.cwd(Directorio) # Cambia Direcotio dentro del FTP
        print ("Cambio de Directorio OK "+Directorio)
        file = open("/home/pi/Documents/ProyectoGrado/FTP/"+Archivo, "rb")
#Ruta y Nombre del Archivo a subir
        print ("Archivo abierto")
        ftp_connection.storbinary('STOR ' + Archivo, file)#Guarda el archivo
en el FTP
        print ("Guardando Archivo..." )
        file.close() #Cierrra el Archivo
        ftp_connection.quit()# Finaliza Conexion con servidr FTP
        print ("Archivo Subido Exitosamente")
        os.remove('/home/pi/Documents/ProyectoGrado/FTP/'+Archivo)#Elimina el
Archivo Local
        print('Archivo Local Borrado')
    except:
        print ("Error Al Subir el Archivo")

#Funcion Toma de Imagenes Periodicas
def CapturaImagen (fecha, hora, fecham, horam):
    camera = PiCamera()
    try:
        fileName='/home/pi/Documents/ProyectoGrado/Fotos/'+fecha+'_'+hora+'.jpg'
        camera.resolution = (2592, 1944)
        camera.start_preview()
        camera.capture(fileName,resize=(520,380))
        camera.stop_preview()
        camera.close()
        # Colocar Fecha y Hora

```

```

        timestampMessage = "Fecha: "+fecham+" Hora: "+horam
        timestampCommand = "/usr/bin/convert " + fileName + " -pointsize 16
\
        -fill yellow -annotate +275+370 '" + timestampMessage + "' " +
fileName
        # Execute our command
        call([timestampCommand], shell=True)
    except:
        camera.stop_preview()
        camera.close()

```

#### F.4 LIBRERÍA DE LECTURA ARCHIVO XML: XMLREAD.PY

A continuación se presenta la librería utilizada para leer el archivo de configuración XML generado por la interfaz gráfica, lo cual permite que los parámetros de configuración ingresados por el usuario puedan ser interpretadas por la aplicación base en python3. Se verifica la activación de cada uno de los canales de entrada y se realiza la lectura de las variables como la identificación de la estación y los diferentes tipos de sensores ya sean análogos, digitales o seriales SDI12.

##### Código 8. Lectura de variables del archivo de configuración XML

```

import xmltodict
archivo =
open('/home/pi/Documents/InterfazDCP_01/InterfazDCP_01/bin/Debug/Configur
ationDCP.xml')
conf = xmltodict.parse(archivo.read())

NombreEstacion=conf['Configuration']['NombreEstacion']
CodigoEstacion=conf['Configuration']['CodigoEstacion']
TiempoTx=float(conf['Configuration']['TiempoTx'])
Muestreo=float(conf['Configuration']['Muestreo'])
ActivaTx=conf['Configuration']['ActivaTx']
SaveDB=conf['Configuration']['SaveDB']

Analogos=[]
Digitales=[]
Seriales=[]

EstadoCH0=conf['Configuration']['EstadoCH0']
if (EstadoCH0=='true'):
    NombreCH0=conf['Configuration']['NombreCH0']
    IDCH0=int(conf['Configuration']['IDCH0'])
    UnidadCH0=conf['Configuration']['UnidadCH0']
    DecimalesCH0=float(conf['Configuration']['DecimalesCH0'])
    InMinCH0=float(conf['Configuration']['InMinCH0'])
    InMaxCH0=float(conf['Configuration']['InMaxCH0'])

```

```

OutMinCH0=float(conf['Configuration']['OutMinCH0'])
OutMaxCH0=float(conf['Configuration']['OutMaxCH0'])

SensorCH0=[0, NombreCH0, IDCH0, UnidadCH0, DecimalesCH0, InMinCH0,
InMaxCH0, OutMinCH0, OutMaxCH0]
Analogos.append(SensorCH0)

EstadoCH1=conf['Configuration']['EstadoCH1']
if (EstadoCH1=='true'):
    NombreCH1=conf['Configuration']['NombreCH1']
    IDCH1=int(conf['Configuration']['IDCH1'])
    UnidadCH1=conf['Configuration']['UnidadCH1']
    DecimalesCH1=float(conf['Configuration']['DecimalesCH1'])
    InMinCH1=float(conf['Configuration']['InMinCH1'])
    InMaxCH1=float(conf['Configuration']['InMaxCH1'])
    OutMinCH1=float(conf['Configuration']['OutMinCH1'])
    OutMaxCH1=float(conf['Configuration']['OutMaxCH1'])

    SensorCH1=[1,NombreCH1, IDCH1, UnidadCH1, DecimalesCH1, InMinCH1,
InMaxCH1, OutMinCH1, OutMaxCH1]
    Analogos.append(SensorCH1)

EstadoCH2=conf['Configuration']['EstadoCH2']
if (EstadoCH2=='true'):
    NombreCH2=conf['Configuration']['NombreCH2']
    IDCH2=int(conf['Configuration']['IDCH2'])
    UnidadCH2=conf['Configuration']['UnidadCH2']
    DecimalesCH2=float(conf['Configuration']['DecimalesCH2'])
    InMinCH2=float(conf['Configuration']['InMinCH2'])
    InMaxCH2=float(conf['Configuration']['InMaxCH2'])
    OutMinCH2=float(conf['Configuration']['OutMinCH2'])
    OutMaxCH2=float(conf['Configuration']['OutMaxCH2'])

    SensorCH2=[2,NombreCH2, IDCH2, UnidadCH2, DecimalesCH2, InMinCH2,
InMaxCH2, OutMinCH2, OutMaxCH2]
    Analogos.append(SensorCH2)

EstadoCH3=conf['Configuration']['EstadoCH3']
if (EstadoCH3=='true'):
    NombreCH3=conf['Configuration']['NombreCH3']
    IDCH3=int(conf['Configuration']['IDCH3'])
    UnidadCH3=conf['Configuration']['UnidadCH3']
    DecimalesCH3=float(conf['Configuration']['DecimalesCH3'])
    InMinCH3=float(conf['Configuration']['InMinCH3'])
    InMaxCH3=float(conf['Configuration']['InMaxCH3'])
    OutMinCH3=float(conf['Configuration']['OutMinCH3'])
    OutMaxCH3=float(conf['Configuration']['OutMaxCH3'])

    SensorCH3=[3,NombreCH3, IDCH3, UnidadCH3, DecimalesCH3, InMinCH3,
InMaxCH3, OutMinCH3, OutMaxCH3]
    Analogos.append(SensorCH3)

EstadoCH4=conf['Configuration']['EstadoCH4']
if (EstadoCH4=='true'):

```

```

NombreCH4=conf['Configuration']['NombreCH4']
IDCH4=int(conf['Configuration']['IDCH4'])
UnidadCH4=conf['Configuration']['UnidadCH4']
DecimalesCH4=float(conf['Configuration']['DecimalesCH4'])
InMinCH4=float(conf['Configuration']['InMinCH4'])
InMaxCH4=float(conf['Configuration']['InMaxCH4'])
OutMinCH4=float(conf['Configuration']['OutMinCH4'])
OutMaxCH4=float(conf['Configuration']['OutMaxCH4'])

SensorCH4=[4,NombreCH4, IDCH4, UnidadCH4, DecimalesCH4, InMinCH4,
InMaxCH4, OutMinCH4, OutMaxCH4]
Analogos.append(SensorCH4)

EstadoCH5=conf['Configuration']['EstadoCH5']
if (EstadoCH5=='true'):
NombreCH5=conf['Configuration']['NombreCH5']
IDCH5=int(conf['Configuration']['IDCH5'])
UnidadCH5=conf['Configuration']['UnidadCH5']
DecimalesCH5=float(conf['Configuration']['DecimalesCH5'])
InMinCH5=float(conf['Configuration']['InMinCH5'])
InMaxCH5=float(conf['Configuration']['InMaxCH5'])
OutMinCH5=float(conf['Configuration']['OutMinCH5'])
OutMaxCH5=float(conf['Configuration']['OutMaxCH5'])

SensorCH5=[5,NombreCH5, IDCH5, UnidadCH5, DecimalesCH5, InMinCH5,
InMaxCH5, OutMinCH5, OutMaxCH5]
Analogos.append(SensorCH5)

EstadoCH6=conf['Configuration']['EstadoCH6']
if (EstadoCH6=='true'):
NombreCH6=conf['Configuration']['NombreCH6']
IDCH6=int(conf['Configuration']['IDCH6'])
UnidadCH6=conf['Configuration']['UnidadCH6']
DecimalesCH6=float(conf['Configuration']['DecimalesCH6'])
InMinCH6=float(conf['Configuration']['InMinCH6'])
InMaxCH6=float(conf['Configuration']['InMaxCH6'])
OutMinCH6=float(conf['Configuration']['OutMinCH6'])
OutMaxCH6=float(conf['Configuration']['OutMaxCH6'])

SensorCH6=[6,NombreCH6, IDCH6, UnidadCH6, DecimalesCH6, InMinCH6,
InMaxCH6, OutMinCH6, OutMaxCH6]
Analogos.append(SensorCH6)

EstadoCH7=conf['Configuration']['EstadoCH7']
if (EstadoCH7=='true'):
NombreCH7=conf['Configuration']['NombreCH7']
IDCH7=int(conf['Configuration']['IDCH7'])
UnidadCH7=conf['Configuration']['UnidadCH7']
DecimalesCH7=float(conf['Configuration']['DecimalesCH7'])
InMinCH7=float(conf['Configuration']['InMinCH7'])
InMaxCH7=float(conf['Configuration']['InMaxCH7'])
OutMinCH7=float(conf['Configuration']['OutMinCH7'])
OutMaxCH7=float(conf['Configuration']['OutMaxCH7'])

```

```

    SensorCH7=[7,NombreCH7, IDCH7, UnidadCH7, DecimalesCH7, InMinCH7,
InMaxCH7, OutMinCH7, OutMaxCH7]
    Analogos.append(SensorCH7)

EstadoCH8=conf['Configuration']['EstadoCH8']
if (EstadoCH8=='true'):
    NombreCH8=conf['Configuration']['NombreCH8']
    IDCH8=int(conf['Configuration']['IDCH8'])
    UnidadCH8=conf['Configuration']['UnidadCH8']
    DecimalesCH8=float(conf['Configuration']['DecimalesCH8'])
    ResCH8=float(conf['Configuration']['ResCH8'])

    SensorCH8=[8,NombreCH8, IDCH8, UnidadCH8, DecimalesCH8, ResCH8]
    Digitales.append(SensorCH8)

EstadoCH9=conf['Configuration']['EstadoCH9']
if (EstadoCH9=='true'):
    NombreCH9=conf['Configuration']['NombreCH9']
    IDCH9=int(conf['Configuration']['IDCH9'])
    UnidadCH9=conf['Configuration']['UnidadCH9']
    DecimalesCH9=float(conf['Configuration']['DecimalesCH9'])
    ResCH9=float(conf['Configuration']['ResCH9'])

    SensorCH9=[9,NombreCH9, IDCH9, UnidadCH9, DecimalesCH9, ResCH9]
    Digitales.append(SensorCH9)

EstadoCH10=conf['Configuration']['EstadoCH10']
if (EstadoCH10=='true'):
    NombreCH10=conf['Configuration']['NombreCH10']
    IDCH10=int(conf['Configuration']['IDCH10'])
    UnidadCH10=conf['Configuration']['UnidadCH10']
    DecimalesCH10=float(conf['Configuration']['DecimalesCH10'])
    SlaveCH10=float(conf['Configuration']['SlaveCH10'])
    PosCH10=float(conf['Configuration']['PosCH10'])

    SensorCH10=[10,NombreCH10, IDCH10, UnidadCH10, DecimalesCH10,
SlaveCH10, PosCH10]
    Seriales.append(SensorCH10)

EstadoCH11=conf['Configuration']['EstadoCH11']
if (EstadoCH11=='true'):
    NombreCH11=conf['Configuration']['NombreCH11']
    IDCH11=int(conf['Configuration']['IDCH11'])
    UnidadCH11=conf['Configuration']['UnidadCH11']
    DecimalesCH11=float(conf['Configuration']['DecimalesCH11'])
    SlaveCH11=float(conf['Configuration']['SlaveCH11'])
    PosCH11=float(conf['Configuration']['PosCH11'])

    SensorCH11=[11,NombreCH11, IDCH11, UnidadCH11, DecimalesCH11,
SlaveCH11, PosCH11]
    Seriales.append(SensorCH11)

```

## F.5 PROGRAMA LECTURA DATOS SDI-12: ARDU\_SDII12\_RX.INO

El código presentado a continuación es el implementado dentro de la plataforma Arduino Nano, con el fin de realizar la comunicación con los sensores que utilizan el protocolo SDI-12. Con la ayuda de la librería especializada **SDI12.h** el programa envía el comando **aM!**, para iniciar la medición y luego de esperar el tiempo de respuesta del sensor, envía el comando **aD0!**, para realizar la lectura de los valores, adicionalmente dependiendo de la posición establecida por el usuario en la configuración del prototipo, el programa devuelve el valor asociado a dicha posición.

### Código 9. Programa en Arduino para la lectura de sensores SDI-12

```
#include "SDI12.h"
#define DATAPIN 10
SDI12 mySDI12 (DATAPIN);

void setup(){
  Serial.begin(9600);
  mySDI12.begin();
}

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length()-1;

  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==separator || i==maxIndex){
      found++;
      strIndex[0] = strIndex[1]+1;
      strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }
  return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void loop(){
  if (Serial.available())
  {
    String cadena = "";
    cadena = Serial.readString();
    String A =cadena.substring(0,1);
    String P =cadena.substring(1,2);

    String ComandaM=A+"M!";
    mySDI12.sendCommand(ComandaM);
    String atttn= "";
    delay(30);
    while(mySDI12.available()){
```

```

    char c = mySDI12.read();
    if ((c != '\n') && (c != '\r'))
    {
        atttn += c;
        delay(5);
    }
}
mySDI12.flush();

int a = 0;
int ttt = 0;
int n = 0;
a = atttn.substring(0,1).toInt();
ttt = atttn.substring(1,4).toInt();
n = atttn.substring(4).toInt();
delay(ttt);

String ComandaD0="A+\"D0!";
mySDI12.sendCommand(ComandaD0);
while(!mySDI12.available()>1);
delay(30);

String buffer = "";
mySDI12.read();
while(mySDI12.available()){
    char c = mySDI12.read();
    if(c == '+' || c == '-'){
        buffer += ',';
        if(c == '-') buffer += '-';
    }
    else if ((c != '\n') && (c != '\r')) {
        buffer += c;
    }
    delay(50);
}
int p = P.toInt();
String lectura = getValue(buffer, ',', p);
Serial.println(lectura);
Serial.flush();
delay(1000);
mySDI12.flush();
}
}

```