



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES.

TEMA:

“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN
METEOROLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL
DE LOJA.”

TESIS DE GRADO PREVIA A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN ELECTRÓNICA
Y TELECOMUNICACIONES.

DIRECTOR:

ING. JUAN PABLO CABRERA SAMANIEGO, Mg. Sc.

AUTOR:

VICENTE RIGOBERTO POMA GODOY

LOJA-ECUADOR

1859
2015

CERTIFICACIÓN

Ing.

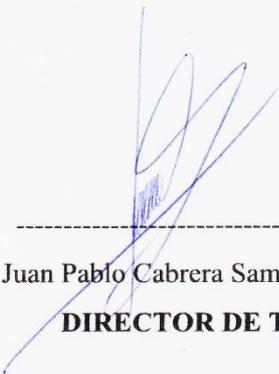
Juan Pablo Cabrera Samaniego, Mg. Sc.

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación, cuyo tema versa; **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA”**, previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el señor egresado: Vicente Rigoberto Poma Godoy, la misma que cumple con el reglamento y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, Diciembre del 2014



Ing. Juan Pablo Cabrera Samaniego, Mg. Sc.
DIRECTOR DE TESIS

AUTORÍA

Vicente Rigoberto Poma Godoy, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional - Biblioteca Virtual.

Autor: Vicente Rigoberto Poma Godoy

A handwritten signature in blue ink, consisting of stylized loops and the name 'VICENTE POMA' written in capital letters.

Firma:

Cédula: 1104141260

Fecha: 23 de febrero del 2015

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

VICENTE RIGOBERTO POMA GODOY, declaro ser autor de la tesis titulada: **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA”**, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la y tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, veintitrés días del mes de febrero del dos mil quince.

Firma: 

Autor: Vicente Rigoberto Poma Godoy

Cedula: 1104141260

Dirección: Loja (Cdla. Esteban Godoy, Calles Sldo. Héctor Pilco y Sldo. Carlos Yuqui.)

Correo electrónico: viche_poma28@hotmail.es

Teléfono: 2547802 Celular: 0997323820

DATOS COMPLEMENTARIOS

Director de tesis: Ing. Juan Pablo Cabrera Samaniego, Mg. Sc.

Tribunal de Grado: Ing. Juan Manuel Galindo Vera, Mg. Sc.

Ing. Yessenia Berenice Llive Barragán, Mg. Sc.

Ing. John Jossimar Tucker Yépez, Mg.Sc.

DEDICATORIA

El presente trabajo va dedicado especialmente a mis queridos padres, hermanos por la incondicional ayuda y fortaleza que me supieron brindar para ser mejor cada día, también va dedicada muy especialmente a mi pequeña hijita Valentina que ha sido un factor fundamental de inspiración para poder terminar mis estudios y formarme como un buen profesional.

AGRADECIMIENTO

Vicente Rigoberto Poma Godoy, autor del presente trabajo de tesis dejo constancia de mi agradecimiento:

A mi Dios y la Virgen del Cisne por haberme guiado por un buen sendero de la vida en los momentos difíciles y por la fortaleza para no desmayar en los problemas que se me presentaren.

A mis padres, hermanos, hermanas por su confianza, esfuerzo y apoyo incondicional para la formación y culminación de mi carrera universitaria.

A la Universidad Nacional de Loja, en especial a la carrera de Ingeniería Electrónica y Telecomunicaciones que a través de sus docentes me brindaron los conocimientos para mi formación profesional.

Mis más sinceros agradecimientos al Ing. Juan Pablo Cabrera Samaniego director de tesis de la presente investigación, así como también a la planta docente que con sus conocimientos nos supieron enseñar e incentivar en el desarrollo del presente trabajo de tesis.

ÍNDICE DE CONTENIDOS

a	TÍTULO	1
b	RESUMEN.....	2
b.1	ABSTRACT.....	3
c	INTRODUCCIÓN	3
d	REVISIÓN LITERARIA	5
d.1	METEOROLOGÍA	5
d.1.1	ESTACIONES METEOROLÓGICAS	5
d.1.2	CARACTERÍSTICAS	5
d.1.2.1	TEMPERATURA.....	6
d.1.2.2	HUMEDAD RELATIVA	6
d.1.2.3	VELOCIDAD DEL VIENTO.....	6
d.1.2.4	DIRECCIÓN DEL VIENTO	6
d.1.2.5	PRECIPITACIÓN.....	6
d.1.3	CLASIFICACIÓN DE ESTACIONES METEOROLÓGICAS	7
d.1.4	ADQUISICIÓN Y ALMACENAMIENTO DE LAS VARIABLES METEOROLÓGICAS	8
d.1.4.1	SENSOR	8
d.1.4.1.1	TERMÓMETRO E HIGRÓMETRO	8
d.1.4.1.2	ANEMÓMETRO Y VELETA	8
d.1.4.1.3	PLUVIÓMETRO.....	9
d.1.4.2	DATALOGGER	9
d.2	REDES PARA TRANSMISIÓN DE DATOS	9
d.2.1	TECNOLOGÍA CELULAR	10
d.2.1.1	RED GSM/GPRS.....	11
d.2.1.2	ARQUITECTURA	13
d.2.1.3	ELEMENTOS DE RED GSM.....	13
d.2.1.4	ELEMENTOS DE RED GPRS	15
d.2.2	PROTOCOLO GPRS.....	17
d.2.2.1	PILA DE PROTOCOLOS DE SM.....	17
d.2.2.2	PROTOCOLO HTTP.....	18
d.2.2.3	PROTOCOLO GTP	19
d.2.2.4	TRAMA GPRS	19
d.2.2.5	MODULACIÓN GPRS	21
d.2.2.6	CODIFICACIÓN GPRS	21
d.2.3	CLASE DE TERMINALES MÓVILES	22
d.2.3.1	CLASE A	22
d.2.3.2	CLASE B	22
d.2.3.3	CLASE C	22
d.2.4	COMANDOS AT	23
d.2.4.1	SINTAXIS DE LOS COMANDOS AT	23
d.2.4.2	COMANDOS.....	24

d.3	GITHUB.....	25
d.3.1	CONTROL DE VERSIONES	25
d.3.1.1	HISTORIA DE GIT	25
d.3.2	FUNDAMENTOS DE GIT	26
d.3.2.1	INSTALACIÓN DE GIT.....	28
d.3.2.2	CONFIGURACIÓN DE GIT	29
d.3.2.3	GENERAR UNA CLAVE PÚBLICA SSH	29
d.3.2.4	CREAR UN REPOSITORIO GIT	30
d.3.2.5	GUARDANDO CAMBIOS EN EL REPOSITORIO	31
d.3.2.6	REPOSITORIOS REMOTOS	31
d.3.2.6.1	CREAR REPOSITORIOS REMOTOS.....	32
d.3.2.6.2	SUBIR EL REPOSITORIO REMOTO.....	33
d.3.2.6.3	GESTIONAR UN REPOSITORIO REMOTO	33
d.3.2.6.4	RENOMBRAR Y ELIMINAR REPOSITORIOS REMOTOS	33
d.3.2.7	HISTORIAL DE LAS CONFIRMACIONES	34
d.3.3	RAMIFICACIONES EN GIT.....	34
d.3.3.1	PROCEDIMIENTOS BÁSICOS DE RAMIFICACIÓN	35
d.3.3.2	PROCEDIMIENTOS BÁSICOS DE FUSIÓN	35
d.3.3.3	GESTIÓN DE RAMIFICACIONES	36
e	MATERIALES Y MÉTODOS	37
e.1	MATERIALES	37
e.1.1	DIAGRAMA DE BLOQUES DE PROTOTIPO IMPLEMENTADO	37
e.1.2	PLATAFORMA ARDUINO	38
e.1.3	ARDUINO UNO (ATMEGA328).....	39
e.1.3.1	VOLTAJE DE ALIMENTACIÓN	39
e.1.3.2	MEMORIA	40
e.1.3.3	ENTRADAS Y SALIDAS	40
e.1.3.4	RELÉ DE PROTECCIÓN DE SOBRECORRIENTE USB.....	41
e.1.3.5	COMUNICACIÓN	41
e.1.3.6	PROGRAMACIÓN	41
e.1.3.7	LIBRERÍAS	41
e.1.4	MÓDULO GSM/GPRS	42
e.1.4.1	DESCRIPCIÓN DEL HARDWARE	43
e.1.4.2	ENCENDIDO POR SOFTWARE.....	44
e.1.5	DHT22	45
e.1.6	ANEMÓMETRO Y VELETA.....	47
e.1.7	PLUVIÓMETRO	49
e.1.8	MÓDULO MICRO SD.....	511
e.1.8.1	MEMORIA SD	53
e.1.9	RELOJ EN TIEMPO REAL (RTC).....	53
e.2	MÉTODOS	55
e.2.1	DISEÑO DEL SOFTWARE.....	55

e.2.1.1	FUCIONAMIENTO DE LOS SENSORES	55
e.2.1.1.1	CALIBRACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD	56
e.2.1.1.2	CALIBRACIÓN DE LOS SENSORES DE VELOCIDAD Y DIRECCIÓN DEL VIENTO	58
e.2.1.1.3	CALIBRACIÓN DEL SENSOR DE PRECIPITACIÓN.....	61
e.2.1.2	DATALOGGER	63
e.2.2	COMUNICACIÓN A INTERNET.....	65
e.2.2.1	CONFIGURACIÓN DE LA RED.....	65
e.2.2.2	CREACIÓN DE UNA INTERFAZ PARA EL REGISTRO DE DATOS.....	67
e.2.3	SISTEMA DE CONTROL DE VERSIONES GITHUB PARA PRESENTACIÓN DE PÁGINA WEB.	74
e.2.3.1	CONFIGURACIÓN EN GITHUB	74
e.2.3.2	MODELO DE PRESENTACIÓN DE LOS DATOS METEOROLÓGICOS	78
f	RESULTADOS	80
f.1	SENSOR DE TEMPERATURA.....	80
f.2	SENSOR DE HUMEDAD.....	81
f.3	SENSOR DE VELOCIDAD DEL VIENTO	81
f.4	SENSOR DE DIRECCIÓN DEL VIENTO.....	82
f.5	SENSOR DE PRECIPITACIÓN	83
f.6	PRUEBAS DE ALMACENAMIENTO DE LOS DATOS EN UNA MEMORIA SD.....	83
f.6.1	ESPECIFICACIONES DE ALMACENAMIENTO	84
f.7	PRUEBAS DE COMUNICACIÓN PARA REGISTRO DE LOS DATOS	84
f.8	CARACTERÍSTICAS TÉCNICAS DEL PROTOTIPO DE ESTACIÓN METEOROLÓGICA.	95
f.9	ANÁLISIS ECONÓMICO Y COMPARATIVO DEL PROTOTIPO DE ESTACIÓN METEOROLÓGICA DISEÑADA, CON UNA COMERCIAL	96
f.9.1	COSTO DEL DISPOSITIVO IMPLEMENTADO	96
f.9.2	COMPARACIÓN DEL PROTOTIPO IMPLEMENTADO CON ESTACIONES METEOROLÓGICAS COMERCIALES	97
g	DISCUSIÓN.....	100
h	CONCLUSIONES	104
i	RECOMENDACIONES	106
j	BIBLIOGRAFÍA.....	108
k	ANEXOS.....	111

ÍNDICE DE FIGURAS

Figura 1: Esquema básico de la telefonía celular. Fuente: (7).....	10
Figura 2: Sistema celular. Fuente: (8).	11
Figura 3: Arquitectura de red GSM/GPRS. Fuente: (9).....	13
Figura 4: Pila de protocolos de una estación móvil. Fuente: (12).....	17
Figura 5: Estructura de trama TDMA. Fuente: (14).....	20
Figura 6: Pasos para codificación. Fuente: (El Autor).....	21
Figura 7: Sistemas que almacenan datos como cambios de cada archivo. Fuente: (15).	27
Figura 8: Git almacena la información como instantáneas. Fuente: (15).	27
Figura 9: Directorio de trabajo, área de preparación y directorio de Git. Fuente: (15).	28
Figura 10: El ciclo de vida del estado de tus archivos. Fuente: (15).....	31
Figura 11: Apuntadores en el registro de confirmaciones de una rama. Fuente: (15). 34	
Figura 12: Un registro de confirmaciones simple y corto. Fuente: (15).....	35
Figura 13: Diagrama de bloques del prototipo de estación meteorológica. Fuente: (El Autor).....	38
Figura 14: Arduino ATmega328. Fuente: (17).....	42
Figura 15: Descripción Shield GSM/GPRS Fuente: (18).....	44
Figura 16: Conexión de hardware para modo encendido por Software. Fuente: (18). 44	
Figura 17: Sistema de encendido por software del módulo SIM900. Fuente: (19).....	45
Figura 18: Sensor DHT22. Fuente: (El Autor).....	45
Figura 19: Esquema de conexión del sensor DHT22. Fuente: (El Autor).	47
Figura 20: Anemómetro y veleta. Fuente: (El Autor).	47
Figura 21: Esquema del Anemómetro y la Veleta. Fuente: (El Autor).	48
Figura 22: Esquema de conexión del anemómetro y veleta. Fuente: (El Autor).	49
Figura 23: Pluviómetro Fuente: (El Autor).....	50
Figura 24: Esquema de conexión del pluviómetro. Fuente: (El Autor).	51
Figura 25: Módulo SD para Arduino. Fuente: (20).	52
Figura 26: Esquema de conexión del Arduino con el módulo SD. Fuente:(El Autor). .	52
Figura 27: Circuito integrado DS1307. Fuente: (El Autor).....	54
Figura 28: Descripción de los pines del DS1307. Fuente: (El Autor).	54
Figura 29: Esquema del RTC. Fuente: (El Autor).....	54
Figura 30: Estructura del software implementado. Fuente: (El Autor).	55
Figura 31: Diagrama de flujo del inicio del algoritmo implementado para el prototipo de estación meteorológica. Fuente: (El Autor).	56
Figura 32: Señal de Start. Fuente: (21).....	57
Figura 33: Diagrama de flujo de la temperatura y humedad. Fuente: (El Autor).	58
Figura 34: Diagrama de flujo de la velocidad del viento. Fuente: (El Autor).	59
Figura 35: Circuito para calibrar la veleta. Fuente: (El Autor).....	60
Figura 36: Diagrama de flujo de la dirección del viento. Fuente: (El Autor).	61

Figura 37: Diagrama de flujo de la intensidad de lluvia. Fuente: (El Autor).	62
Figura 38: Diagrama de flujo para el almacenamiento de los datos generados por los sensores. Fuente: (El Autor).	64
Figura 39: Diagrama de flujo de la conexión a la red de internet. Fuente: (El Autor).	66
Figura 40: Entorno de las entradas de emoncms.org. Fuente: (22).	68
Figura 41: Entorno del campo "Feed" de emoncms. Fuente: (22).	69
Figura 42: Visualización de variables en emoncms.org. Fuente: (El Autor).	70
Figura 43: Interfaz gráfica para monitorear temperatura, humedad, en tiempo real. Fuente: (El Autor).	72
Figura 44: Interfaz gráfica para el monitoreo de la temperatura y la humedad. Fuente: (El Autor).	72
Figura 45: Interfaz gráfica para el monitoreo de la Velocidad y la dirección del viento en tiempo real. Fuente: (El Autor).	73
Figura 46: Interfaz gráfica para el monitoreo de la precipitación. Fuente: (El Autor).	73
Figura 47: Interfaz gráfica para la configuración de la cuenta en GitHu. Fuente: (El Autor).	75
Figura 48: Interfaz gráfica para la configuración de la cuenta en GitHub. Fuente: (El Autor).	75
Figura 49: Interfaz gráfica para la creación de un repositorio en GitHub. Fuente: (El Autor).	76
Figura 50: Interfaz gráfica de la consola git bash para la configuración de cuenta en GitHub. Fuente: (El Autor).	77
Figura 51: Presentación de archivos subidos en GitHub a través de la consola git Bash. Fuente: (El Autor).	78
Figura 52: Imagen de la página principal del prototipo de estación meteorológica. Fuente: (El Autor).	79
Figura 53: Gráfica de los datos de cada una de las variables sensadas en el servidor emoncms.org. Fuente: (El Autor).	86
Figura 54: Gráfica de inicio en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	87
Figura 55: Gráfica de proyecto en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	88
Figura 56: Gráfica de temperatura en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	90
Figura 57: Gráfica de velocidad y dirección del viento en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	92
Figura 58: Gráfica de precipitación pluvial en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	93
Figura 59: Gráfica de galería en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).	94

Figura 60: Esquema de conexión utilizado para la adquisición de las variables climatológicas, simulado en Fritzing. Fuente: (El Autor).	111
Figura 61: Esquema de conexión utilizado para la transmisión de los datos, del prototipo de estación meteorológica, simulado en Fritzing. Fuente: (El Autor).	112
Figura 62: Diseño del circuito impreso para el acople de los distintos sensores utilizados, realizado en ARES. Fuente: (El Autor).	112
Figura 63: Gráfica de la placa impresa para la adecuación de RTC, SD Y SENSORES. Fuente: (El Autor).	113
Figura 64: Gráfica de los sensores: DHT22, anemómetro, veleta y pluviómetro. Fuente: (El Autor).	113
Figura 65: Gráfica del datalogger y la adecuación para los sensores.. Fuente: (El Autor).	114
Figura 66: Gráfica del prototipo implementado. Fuente: (El Autor).	114
Figura 67: Indicaciones técnicas del prototipo de estación meteorológica. Fuente: (El Autor).	115

ÍNDICE DE TABLAS

Tabla 1: Clasificación de estaciones meteorológicas. Fuente: (4).....	7
Tabla 2: Tipos de codificación de datos GPRS. Fuente: (11).....	22
Tabla 3: Principales características de Arduino Uno. Fuente: (3).	42
Tabla 4: Características técnicas de un DHT22. Fuente: (El Autor).	46
Tabla 5: Características técnicas del anemómetro y la veleta. Fuente: (El Autor)	48
Tabla 6: Características técnicas del pluviometro. Fuente: (El Autor).....	51
Tabla 7: Presentación de los datos de cada una de las variables almacenadas en la SD. Fuente: (El Autor).	64
Tabla 8: Comparación de las medidas de temperatura. Fuente: (El Autor).	81
Tabla 9: Medidas de humedad relativa. Fuente: (El Autor).	81
Tabla 10: Comparación de las medidas de la velocidad del viento. Fuente: (El Autor).	82
Tabla 11: Comparación de las medidas de la dirección del viento. Fuente: (El Autor).	82
Tabla 13: Comparación de las medidas de precipitación. Fuente: (El Autor).....	83
Tabla 14: Registro de las variables del prototipo de estación meteorológica en una memoria externa. Fuente: (El Autor).	84
Tabla 14: Características del prototipo de estación meteorológica. Fuente: (El Autor).	95
Tabla 15: Presupuesto del hardware para la estación meteorológica. Fuente: (El Autor).	96
Tabla 16: Comparación del prototipo diseñado con las estaciones meteorológicas comerciales. Fuente: (El Autor).	97
Tabla 17: Tabla de datos almacenados en la SD del prototipo de estación meteorológica. Fuente: (El Autor).	117

PENSAMIENTO

“Nunca consideres el estudio como un deber, sino como una oportunidad para penetrar en el maravilloso mundo del saber”

Albert Einstein

ACRÓNIMO	DEFINICIÓN
AC	Corriente Alterna.
ADC	Convertidor Analógico Digital.
AREF	Voltaje de Referencia Externa.
ASCII	Código Estándar Estadounidense para el Intercambio de Información.
ATM	Modo De Transferencia Asíncrona.
AUC	Centro de Autenticación.
BSC	Controlador de Estaciones Base.
BSS	Sistema de Estación Base.
BTS	Transceptor de Estación Base.
CDR	Registro de Detalle de Llamada.
CEPT	Conferencia Europea de Administraciones de Correos y Telecomunicaciones.
CG	Gateway de Carga.
CLNP	Protocolo de Red Sin Conexión.
CVS	Sistema de Control de Versiones.
DC	Corriente Directa.
DHT	Sensor de Temperatura y Humedad Digital.
EEPROM	Memoria Solo Lectura Programable y Borrable Eléctricamente.
EIR	Registro de Identidad del Equipo.
EPROM	Memoria Solo de Lectura Programable y Borrable.

ETSI	Instituto de Estándares de Telecomunicaciones Europeo.
FTP	Protocolo de Transferencia de Archivos.
GGSN	Nodo de Soporte Gateway GPRS.
GMSC	Centro de Conmutación de Servicios Móviles Gateway.
GPIO	Entrada y Salida de Propósito General.
GPRS	Servicio General de Paquetes Vía Radio.
GSM	Sistema Global de Comunicaciones Móviles.
GSM RF	Radio Frecuencia de GSM.
GSN	Nodo de Soporte GPRS.
GTP	Protocolo Tunelling GPRS.
HLR	Registro Local del Abonado.
HTTP	Protocolo de Transferencia de Hipertexto.
I2C	Inter-Circuitos Integrados (Comunicación Serie).
ICSP	Programación Serie en Circuitos.
INAMHI	Instituto Nacional de Meteorología e Hidrología.
IOREF	Entradas y Salida de Referencia.
ISP	Proveedor de Servicios de Internet.
LLC	Control de Enlace Lógico.
MAC	Control de Acceso al Medio.
MISO	Salida Esclavo Entrada Master.
MMS	Sistema de Mensajería Multimedia.
MOSI	Entrada Esclavo Salida Master.

MS	Estación Móvil.
MSC	Centro de Conmutación de Servicios Móviles.
NSS	Subsistema de Red y Conmutación.
OMM	Organización Meteorológica Mundial.
PWM	Modulación por Ancho de Pulso.
RDSI	Red Digital de Servicios Integrados.
RLC	Control de Enlace de Radio.
RSS	Subsistema de Radio.
RTC	Reloj en Tiempo Real.
RX	Recepción.
SAP	Servicio de Punto de Acceso.
SCL	Línea de Reloj.
SDA	Línea de Datos.
SGSN	Nodo de Soporte de Servicios GPRS.
SIM	Módulo de Identificación de Abonado.
SIMCOM	Módulos Inalámbricos Basados en GSM/GPRS.
SMS	Servicio de Mensajería Corto.
SNDCP	Sub Red de Protocolo de Convergencia Dependiente.
SPI	Bus Serial de Interfaz de Periféricos.
SRAM	Memoria Estática de Acceso Aleatorio.
SSH	Intérprete de Órdenes Segura.
TCP	Protocolo de Control de Transmisión.

TDMA	Acceso Múltiple por División de Tiempo.
TTL	Lógica Transistor a Transistor.
TWI	Dos Hilos (To wire).
TX	Transmisión.
UART	Transmisor-Receptor Asíncrono Universal.
UDP	Protocolo de Datagrama de Usuario.
USB	Bus Universal Serial.
USF	Frecuencia del Lado Superior.
VLR	Registro Temporal de Abonado.

a TÍTULO

“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.”

b RESUMEN

El desarrollo del presente trabajo de investigación está enfocado a medir variables climáticas, mediante un tratamiento adecuado de la señal y una respectiva transmisión, se llegue a presentar los datos en la web, alojándolos en un servidor gratuito, logrando de esta manera cumplir con el objetivo de diseñar y construir un prototipo de una estación meteorológica con monitoreo remoto, en el Área de la Energía, las Industrias y los Recursos Naturales No Renovables de la Universidad Nacional de Loja, siendo de total beneficio para obtener la información del estado climático del sector en diferentes lugares donde nos encontremos.

Para la implementación del prototipo de estación meteorológica podemos detallar en cuatro partes fundamentales como son: la primera es la adquisición de las señales mediante sensores climatológicos (DHT22, anemómetro, veleta, pluviómetro), la segunda es la tarjeta que adecuará las señales de estos sensores y donde también constará de un circuito adicional para el almacenamiento de los datos obtenidos en una micro SD, los mismos que servirán de respaldo en caso de que existirá problemas con la transmisión o algún otro problema que obstruya el envío de los datos, la tercera parte está compuesta por una placa Arduino UNO R3 para el procesamiento digital de los datos y envío de los mismos a través del módulo SIM 900 GSM/GPRS el cual hace la conexión entre la placa anterior mencionada y el servidor emoncms.org y la cuarta parte es la interfaz web tanto del servidor como de la página donde se muestran los datos obtenidos por los sensores.

El diseño del software versa en el desarrollo de un código entendible por el microcontrolador de la placa Arduino, el mismo que tiene sus distintas etapas en donde se puede calcular, calibrar y mediante una instrucción guardar los datos que obtienen los sensores para luego procesar y enviar los mismos a través de una red celular para así llegar a su destino final como es el servidor web emoncms.org.

Finalmente se verificó el funcionamiento del prototipo, determinado de esta manera la funcionalidad, utilidad y factibilidad del proyecto, proceso que se lo realizó en base a las pruebas y análisis comparativo con otras estaciones meteorológicas.

b.1 ABSTRACT

The development of this research work is focused on measuring climatic variables, by means of a proper treatment of the signal and a respective transmission, The data in the web were shown ,setting them in a free server, in this way the objective to design and built a prototype of a meteorological station with remote monitoring was achieved, in the Area of energy Industries and nonrenewable resources of the “ Nacional de Loja “ University, It presents a benefit to obtain information of the climate State of the sector in different places where we are .

The implementation of the weather station prototype can be detailed in four important parts such as: the first is the acquisition of the signals by means of weather sensors (DHT22, wind vane, anemometer, pluviometer), the second is the card that suit the signals from these sensors and where there also will be an additional circuit for the storage of the data obtained in a micro SD which will serve as a backup in case of having problems with the transmission or any other problem that obstructs the sending data, the third part is composed of a ONE R3 Arduino plate for the digital processing of data and the sending of them through the SIM 900 GSM/GPRS module which makes the connection between the mentioned plate before and the emoncms.org server and the fourth part is the web interface Server as the page where the data obtained by the sensors is shown.

The software design is developing based on an understandable code for the microcontroller of the Arduino plate, it has different stages where you can calculate, calibrate and trough an instruction can save data obtained by the sensors , after that it is then processed and sent via cellular network to reach its final destination as it is the web server emoncms.org.

Finally, the functionality of the prototype is verified, determining in this way the utility and feasibility of the project, a process that was made on the basis of the evidence and comparative analysis with other meteorological stations.

c INTRODUCCIÓN

Los fenómenos naturales se manifiestan de diferente forma, así mismo estos influyen de distintas maneras sobre los seres vivos. De tal forma que se hace necesario contar con un registro de los diferentes factores climáticos como: temperatura, humedad, presión atmosférica, velocidad y dirección del viento, precipitación entre otros, así también tener en cuenta las diferentes fluctuaciones que se producen en las distintas épocas del año, ya que estos datos servirán de apoyo para la producción de diferentes áreas.

La influencia de estos factores climáticos ha hecho que se cree estaciones meteorológicas que serán las encargadas de registrar los datos del estado del ambiente, en la actualidad son de vital importancia para diferentes sectores como: agrícolas, ganaderas, geológicos, predicciones climáticas, industriales, también en el área de la generación de energías renovables, lo cual ha tomado importancia en nuestro país por los proyectos que se vienen desarrollando, por lo que se ha creído necesario disponer de una estación meteorológica que provea información climática, con la capacidad para mostrar la evolución de cada una de las variables monitoreadas en la web, de la misma manera proporcionar un registro de las mediciones, lo que contribuirá para una adecuada investigación o el desarrollo de futuros proyectos en los diferentes campos de su competencia.

El poder tecnológico alcanzado en las últimas décadas ha logrado construir gran cantidad de dispositivos elementales para monitorear y transmitir datos a través de una red, para presentarles en diferentes lugares utilizando una red masiva como el internet; que permitirá tener la información necesaria sin tener que movilizarse al lugar donde se encuentre la estación meteorológica, brindando de esta manera grandes ventajas en diferentes áreas de estudio.

En el Ecuador la institución encargada de sensor las diferentes variables ambientales es el Instituto Nacional de Meteorología e Hidrología (INAMHI), el mismo que se encarga de monitorear y dar la información del estado climatológico, de manera constante, en su sitio web, la que se encuentra disponible en diferentes intervalos de tiempo (horario).

d REVISIÓN LITERARIA

d.1 METEOROLOGÍA

La Meteorología es la ciencia encargada del estudio de la atmósfera, de sus propiedades y los fenómenos que en ella tienen lugar. El estudio de la atmósfera se basa en el conocimiento de una serie de magnitudes o variables meteorológicas como: temperatura, presión atmosférica, humedad y cambios que se producen por fenómenos del viento las cuales varían tanto en el espacio como en el tiempo. (1)

Desde hace mucho tiempo se ha observado los fenómenos atmosféricos intentando explicar las causas que lo producen, incluso cuando aún no existía instrumentos con los que se pudiera obtener datos de los fenómenos que se producían, pero hoy en día existen estaciones meteorológicas que recogen datos los mismos que en manos de un meteorólogo se transforman en un fuente muy valiosa de información.

d.1.1 ESTACIONES METEOROLÓGICAS

Se puede definir una estación meteorológica como una instalación diseñada para medir y registrar regularmente diversas variables climáticas, cuentan en la actualidad con estaciones automáticas las mismas que tienen sensores electrónicos encargados de medir el estado físico del medio ambiente, también pueden tener un sistema de transporte y presentación de los datos. Además suministran toda la información meteorológica especializada en tiempo real, para las actividades como de respuesta inmediata en casos de emergencia ambiental, cabe recalcar que una estación automática tiene la capacidad de vigilar permanentemente y comunicar de inmediato el desarrollo y/o comportamiento del estado de las variables meteorológicas consideradas peligrosas para la vida terrestre.

d.1.2 CARACTERÍSTICAS

Una estación meteorológica, tiene diferentes variables climatológicas a medir, que pueden ser: presión atmosférica, temperatura, humedad, dirección y velocidad del viento, precipitación, radiación solar, visibilidad, entre otras; siendo estas algunas de sus características; a continuación describiremos algunas de carácter importante:

d.1.2.1 TEMPERATURA

La temperatura “es una magnitud variable que depende de la velocidad de las moléculas del aire” (2), siendo estas moléculas el oxígeno y el nitrógeno, las que dan una impresión de calor o frío. La temperatura se expresa en unidades de grados Celsius o Fahrenheit.

d.1.2.2 HUMEDAD RELATIVA

La humedad es la cantidad de vapor de agua que contiene el aire. “Esta cantidad no es constata, si no que dependerá de diversos factores, como si ha llovido recientemente, si estamos cerca del mar, si hay plantas etc.” (2). Entonces la humedad relativa es la relación entre la masa de vapor de agua que tiene una determinada masa de aire y la que tendría si estuviese saturada en la misma temperatura. Esta relación se expresa en porcentaje (%).

d.1.2.3 VELOCIDAD DEL VIENTO

La velocidad del viento se define como la distancia recorrida por una masa de aire en una unidad de tiempo (segundos, minutos, horas, etc.) (3), destacando que el viento se mueve en sentido horizontal con la superficie terrestre. El resultado de la velocidad se expresará en unidades de kilómetros por hora (Km/h) o metros por segundo (m/s), etc.

d.1.2.4 DIRECCIÓN DEL VIENTO

La dirección del viento es de donde proviene o sopla el viento, ubicándolo desde una posición cardinal, de las cuales las principales son: Norte (N), Sur (S), Este (E) y Oeste (W); su unidad de medida es expresada comúnmente en grados sexagesimales o centesimales.

d.1.2.5 PRECIPITACIÓN

Es uno de los fenómenos atmosféricos más notables ya que mide la cantidad de agua líquida, sólida (nieve, granizo) o la mezcla de ambas, que se acumula en un determinado recipiente debidamente graduado en una unidad de medida en litros o mililitros.

d.1.3 CLASIFICACIÓN DE ESTACIONES METEOROLÓGICAS

Las estaciones meteorológicas, de acuerdo a lo establecido por la Organización Mundial de Meteorología (OMM), se clasifican de la siguiente manera:

Tabla 1: Clasificación de estaciones meteorológicas. Fuente: (4)

SEGÚN SU FINALIDAD	CLASIFICACIÓN
Sinóptica	Climatológicas
	Agrícolas
	Especiales
	Aeronáuticas
	Satélites
De acuerdo a la magnitud de las observaciones	Principales
	Ordinarias
	Auxiliares
Por el nivel de observación	Superficie
	Altitud
Según el lugar de observación	Terrestres
	Aéreas
	Marítimas

En base que la meteorología es una ciencia bastante amplia, basada en el estudio de los fenómenos ambientales o climatológicos que se producen todo el tiempo en la atmosfera, son muy utilizados para predecir, las condiciones climáticas favorables para diversos sectores como son: la agricultura, ganadería, implementación de proyectos de energías renovables, entre otros; para toda esta diversidad de sectores existen diferentes tipos de estaciones meteorológicas, incluso se implementan sensores adecuados para lo que se desee medir, y así llevar a cabo un proyecto, por lo que se cree de gran importancia realizar estudios de meteorología para la implementación de proyectos en áreas que lo requieran.

d.1.4 ADQUISICIÓN Y ALMACENAMIENTO DE LAS VARIABLES METEOROLÓGICAS

Para la adquisición de las variables meteorológicas, se utilizará sensores los mismos que después de una correspondiente calibración recogerán los datos pertinentes al estado del clima.

Una vez adquiridas las mediciones de las variables meteorológicas es necesario tener en cuenta su almacenamiento para tener un registro de estas. Para el almacenamiento se debe crear un datalogger que realice esta función. A continuación describiremos algunos de los dispositivos que se implementan en las estaciones meteorológicas para cumplir su función:

d.1.4.1 SENSOR

Un sensor es un dispositivo que detecta o mide manifestaciones de cualidades o fenómenos físicos, químicos; como por ejemplo: la energía, velocidad, tamaño y cantidad (5). Este dispositivo es capaz de transformar la magnitud que se requiera medir, en otra que facilite su medida. Dentro de los cuales destacaremos los más importantes para el desarrollo de este proyecto:

d.1.4.1.1 TERMÓMETRO E HIGRÓMETRO

El termómetro es un instrumento para medir la temperatura, debido al desarrollo tecnológico, en la actualidad se cuenta con termómetros electrónicos digitales, como el sensor DHT22, el mismo que tiene la capacidad de medir la temperatura y humedad relativa al aire. “Está compuesto de un sensor de humedad capacitivo y un termistor para medir el aire circundante y da una señal digital en el pin de datos” (3).

d.1.4.1.2 ANEMÓMETRO Y VELETA

Un anemómetro es un instrumento meteorológico que sirve para medir la velocidad de viento. Así mismo es de gran importancia en el área de las aeronaves y aerogeneradores.

Un anemómetro ubicado en la superficie terrestre, medirá la velocidad del viento reinante. Este aparato está compuesto por copas que giran alrededor de un eje vertical las mismas que con ayuda de un circuito electrónico, se puede obtener la velocidad a la que

se mueve el viento. Su unidad de medida es milla por hora, kilómetro por hora Km/h o metros por segundos (m/s).

La veleta se utiliza para determinar la dirección en que se mueve el viento y está puede estar integrada al anemómetro o puede encontrarse por separado, pero cumpliendo su función.

d.1.4.1.3 PLUVIÓMETRO

El pluviómetro es un instrumento que se emplea en las estaciones meteorológicas para recoger y medir la precipitación pluvial o también denominada como la cantidad de agua caída. Su unidad de medida se expresa en milímetros (mm) de altura y su diseño básico consiste en una abertura superior de entrada de agua al recipiente, que luego es dirigida a través de un embudo hacia un balancín donde se recoge ciertas cantidades de agua, para posteriormente interpretar su medida.

d.1.4.2 DATALOGGER

Un datalogger es un dispositivo electrónico que registra mediciones ordenadas en el tiempo, provenientes de diferentes sensores, luego de cada medición el dato es almacenado junto con su respectiva hora y fecha, permitiendo periodos de muestreo y almacenamiento configurable (6).

En una estación automática un datalogger es un respaldo de los datos ya que estas tiene la posibilidad de poder transferir los mismos desde el lugar de procesamiento hasta una central de monitoreo a través del internet ya sea usando radio enlaces, redes celulares o redes satelitales, los que en cualquier momento pueden sufrir un inconveniente inesperado y perder conectividad.

d.2 REDES PARA TRANSMISIÓN DE DATOS

Las redes de transmisión de datos ha permitido la comunicación desde hace aproximadamente los años 70, donde se empezó a considerar la interconexión entre computadoras y dispositivos, dando como consecuencia en aquella época distintos grados de dificultad para su conexión y cumpliendo el objetivo de que se compartiera recursos entre distintos ordenadores.

En la actualidad las redes son las mayores promotoras de los adelantos tecnológicos ya que a través de estas se puede obtener información de cualquier clase, una muestra de esto es la red mundialmente conocida como es el internet, siendo esta de gran accesibilidad, ya que hoy en día sus factores como: el precio se han reducido considerablemente y su estructura de servicio ha crecido favorablemente ayudando así al desarrollo personal y profesional de los usuarios.

d.2.1 TECNOLOGÍA CELULAR

Los diferentes tipos de dispositivos que se encuentran en el mercado hoy por hoy hacen que se necesite de redes inalámbricas para la constante comunicación por lo que existen algunas organizaciones que han normando este tipo de comunicaciones, entre ellas podemos nombrar a la UIT que aporta normas para la regularización de los servicios de radiotelefonía móvil a nivel mundial.

La tecnología celular nos ha venido demostrando que tiene un futuro prometedor, aunque no se espera que remplace a las redes de comunicación existentes, si bien es cierto las redes cableadas tienen la capacidad de ofrecer velocidades de transmisión mayores que las alcanzadas con las redes inalámbricas.

En la figura 1 se muestra el esquema básico de telefonía celular donde muestra la facilidad con la que se puede movilizarse y permanecer conectado a la red.

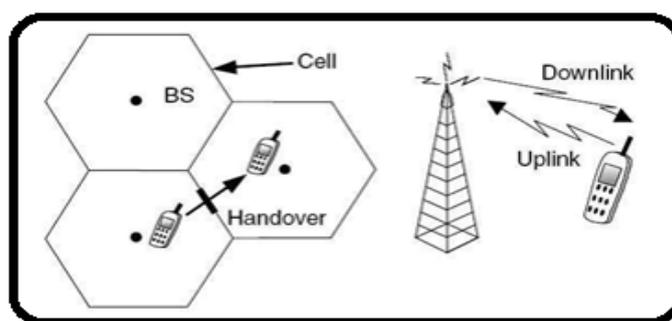


Figura 1: Esquema básico de la telefonía celular. Fuente: (7).

Los sistemas de telefonía móvil, ofrece una gran capacidad de usuarios en áreas limitadas de espectro, la idea fue dividir áreas grandes de cobertura en áreas más pequeñas denominadas células. En la figura 2 se puede observar la división en células para llegar así a los usuarios o llamados también estaciones móviles; en las estaciones base se les

asigna una cierta cantidad de celdas disponibles del número total de un sistema con sus respectivos canales, los mismos que deberán ser distintos a los canales de la estación vecina y de esta manera poder separarlas sistemáticamente, con el objetivo de que se pueda hacer una reutilización de canales cada cierta distancia entre estaciones base y no producir las denominadas interferencias Cocanal¹.

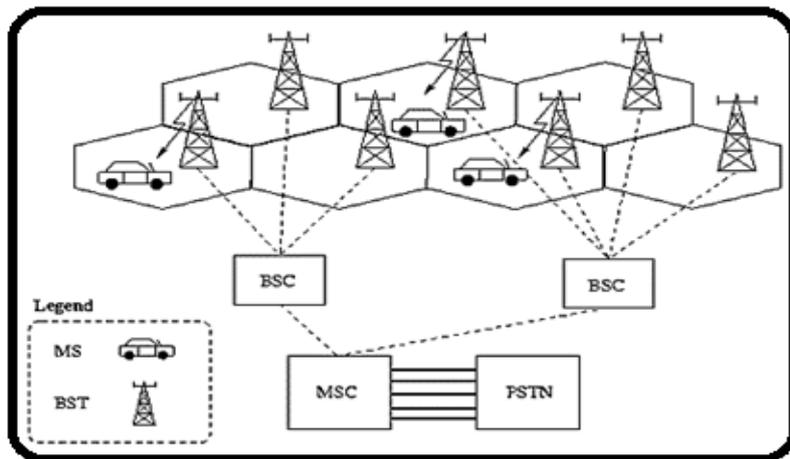


Figura 2: Sistema celular. Fuente: (8).

d.2.1.1 RED GSM/GPRS

La red GSM conocida como Sistema Global para las Comunicaciones Móviles, es un estándar mundial para teléfonos móviles desarrollado en Europa, creada por Conferencia Europea de Administraciones de Correos y Telecomunicaciones (CEPT) y normado posteriormente por el Instituto de Estándares de Telecomunicaciones Europeo (ETSI).

GSM en el Ecuador trabaja en cuatro bandas de frecuencia que son: 850Mhz, 900Mhz, 1800Mhz y 1900Mhz, sus primeros inicios se dieron en las bandas de 850 y 900 MHz, pero tras su rápida expansión, pronto se saturó el espacio radioeléctrico dentro del entorno a esa frecuencia, por lo que se les asignó las frecuencias de 1800 y 1900 MHz; los equipos móviles actuales traen incorporados estos cuatro rangos de frecuencia con los que opera normalmente la telefonía móvil. Con el ingreso de las redes celulares GSM trajo consigo muchas ventajas y amplio desarrollo tecnológico como económico con respecto a sus antecesores (CDMA, TDMA) dentro de las cuales se puede nombrar los siguientes:

¹ Se presenta entre dos BS cercanas que se encuentren transmitiendo a la misma frecuencia, lo que producirá interferencia. En sistemas GSM utilizan saltos de frecuencia para contrarrestar este problema.

- ✓ Transmisión de datos con distintas velocidades binarias.
- ✓ Interconexión con la Red Digital de Servicios Integrados (RDSI).
- ✓ Mejora la seguridad.
- ✓ Mejoras en la calidad de servicio.
- ✓ Mayor calidad en presencia de interferencias: Reducción de la distancia de reuso de frecuencias.
- ✓ Mayor eficacia de las baterías de los portátiles.
- ✓ Terminales y sistema de menor coste.
- ✓ Permite roaming internacional.

Tiempo después nace el sistema de Servicio General de Paquetes Vía Radio (GPRS), que se fundamentó en el mejoramiento de la comunicación móvil GSM para transmitir datos a través de las redes celulares, está es la denominada tecnología 2.5G, la que también brinda servicios de roaming internacional para la transmisión de los datos fuera del país en donde se encuentre operando de forma normal. Esta tecnología trasmite datos de manera inalámbrica, a velocidades máximas de 115 kbps. Este diseño es muy eficiente ya que se caracteriza por proveer una conexión "siempre activa", no exige que el usuario deba conectarse cada vez que desea obtener acceso a datos.

Ventajas de GPRS

- ✓ GPRS se desarrolla sobre la plataforma GSM.
- ✓ GPRS está basado en IP.
- ✓ GPRS utiliza una tecnología abierta y totalmente normalizada, que lo convierte en el ideal para la provisión de acceso inalámbrico a otras redes basadas en IP, tales como LANs corporativas e ISPs.
- ✓ Es una tecnología inalámbrica de paquetes de datos que ofrecen los operadores móviles.
- ✓ Los canales de comunicación se comparten entre los distintos usuarios de manera dinámica, de modo que a un usuario se le asigna un canal dirigido cuando está procesando datos.
- ✓ El sistema de tarifación de GPRS únicamente se realizará cuando existe tráfico de datos.

Características de GPRS

- ✓ Velocidad de transferencia de hasta 115 Kbps.
- ✓ Conexión permanente. Tiempo de establecimiento de conexión inferior al segundo.
- ✓ Pago por cantidad de información transmitida, y mas no por tiempo de conexión.

El servicio de transmisión de datos que posee la red GSM es de gran utilidad para el desarrollo del presente proyecto, que se realiza a través de una red inalámbrica celular que puede proveer de datos a redes basadas en IP, ya que GPRS también se encuentra basada en IP.

d.2.1.2 ARQUITECTURA

La arquitectura no es más que la forma jerárquica que toma la red compuesta por distintos elementos que se muestran en la siguiente figura, donde cada uno cumple una función.

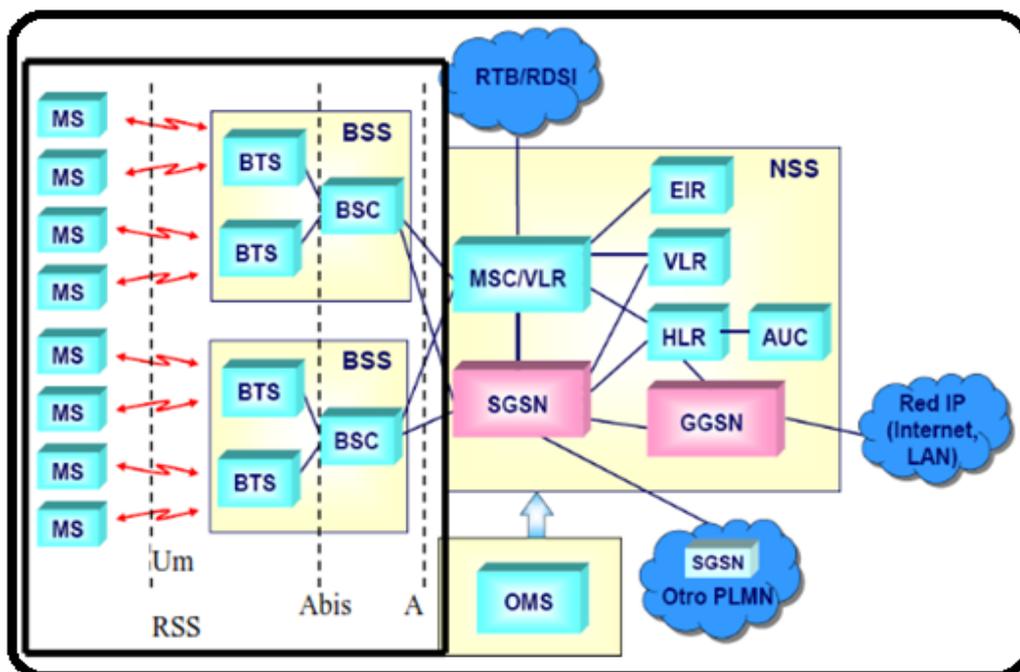


Figura 3: Arquitectura de red GSM/GPRS. Fuente: (9).

d.2.1.3 ELEMENTOS DE RED GSM

A continuación se describirán cada uno de los elementos que integran la arquitectura de la red GSM:

1. **Estación Móvil:** Consta de dos elementos básicos que debemos conocer, por un lado el terminal o equipo móvil y por otro lado el SIM o módulo identificativo del suscriptor. La tarjeta SIM sirve para identificar las características de nuestro terminal. Esta tarjeta se inserta en el móvil y permite al usuario acceder a todos los servicios disponibles por su operador, también tiene el número telefónico que nos asigne una operadora.

La ventaja más importante de la tarjeta SIM es que proporcionan movilidad al usuario ya que puede cambiar de terminal y llevarse consigo.

2. **Subsistema de Radio (RSS):** Cubre la comunicación entre las estaciones móviles (**MS**) y las estaciones base (**BSS**). La interfaz de radio entre estas estaciones se denomina **Um**.

3. **Estación Base (BSS):** es el intermediador para la conexión entre las MS y los NSS, este bloque se encuentra constituido por los siguientes elementos:

- a. **Transceptor de Estación Base (BTS):** Está compuesta por transceiver² y antenas se encuentran en cada célula de la red que sirven como emisor, receptor, con una determinada potencia que será la que determine el tamaño de la célula.

- b. **Controlador de Estaciones Base (BSC):** Se las utiliza como controles de las **BTS**, mapeo de canales de radio sobre los canales terrestres. Como funciones tiene a cargo los handovers³ y las interferencias cocanal. Por un lado se comunica con las **BTS** a través de una interfaz **ABis**⁴ con canales de 16kbts/s y por el otro se comunica con los **MSC** a través de la interfaz “A” con canales de 64 kbts/s.

4. **Subsistema de Red y Conmutación (NSS):** Es un sistema que se encarga de la conmutación, gestión de la movilidad, interconexión con otras redes y control del sistema. Está compuesta por varios sistemas cada uno de ellos con una misión encomendada dentro de la red a continuación los describiremos:

² Dispositivo que hace la función de un transmisor y receptor a la vez.

³ Es el traspaso de un servicio de una estación a otra cuando la calidad del servicio es insuficiente en una de ellas.

⁴ Interfaz utilizada en enlaces de transmisión y señalización de telefonía móvil GSM que funciona a velocidades de 2 Mbit/s según la recomendación G.703 de la UIT-T.

- a. **Centro de Conmutación de Servicios Móviles (MSC):** Son centrales de conmutación que realiza labores dentro de la misma red y también realiza conexiones con otras centrales.
 - b. **Centro de Conmutación de Servicios Móviles Gateway (GMSC):** Es un dispositivo que hace la función de interconectar dos redes diferentes haciendo que los protocolos que existen en ambas redes se conciben.
 - c. **Registro local de abonado (HLR):** Contiene información sobre los abonados conectados a un determinado MSC. La información obtenida es almacenada por ejemplo ID del abonado, número, localización, etc.
 - d. **Registro temporal de abonado (VLR):** Contiene información suficiente para que este pueda acceder a los servicios de dicha red.
 - e. **Centro de autenticación (AUC):** Contiene los parámetros útiles para identificar un usuario en una red, también posee la capacidad para soportar funciones de encriptación.
 - f. **Registro de identidad del equipo (EIR):** Tiene una base de datos con todos los terminales que son válidos para ser usados en la red. En esta base de datos contiene los IMEI de cada terminal.
5. **Red Digital de Servicios Integrados:** Esta red facilita conexiones digitales extremo a extremo proporcionando así una amplia gama de servicios, como por ejemplo los servicios portadores, teleservicios, servicios suplementarios.

d.2.1.4 ELEMENTOS DE RED GPRS

El nodo de soporte GPRS (GSN) es el elemento principal de la infraestructura. Existen dos tipos de nodos GSN: unos de entrada SGSN, que proporcionan conectividad a las BSC de GSM y otros de salida GGSN, que interconectan el sistema con redes de datos externas (10).

Nodo de soporte de servicios GPRS (SGSN).- es el principal componente de una red GPRS. Entre sus funciones podemos nombrar las siguientes:

- ✓ Retransmisión de los datos entre el terminal GPRS y el SGSN.
- ✓ Gestionar la autenticación de los terminales móviles GPRS, si este proceso se completa con éxito, se encarga de su registro en la red GPRS y de su gestión de movilidad.

- ✓ Aviso (Paging). Este procedimiento lo inicia el SGSN para que el terminal móvil pase del estado STANDBY al READY con objeto de poder llevar a cabo el intercambio de datos.
- ✓ Recoger datos necesarios para generar CDRs (Call Detail Recordings) de facturación y enviarlos al CG⁵.
- ✓ Gestionar la conversión del protocolo IP empleado en la red troncal a los protocolos SMDCP y LLC empleados entre el SGSN y el terminal móvil.

Nodo de soporte Gateway (GGSN).- Este proporciona la interconexión entre la red GPRS y las redes de paquetes de datos externas, como por ejemplo Internet, Intranets corporativas, etc. Entre las funciones principales tenemos:

- ✓ Recibir datos de usuario desde una Intranet o Internet y enviarlos hacia el SGSN que controla el terminal a través de la red troncal mediante el protocolo de tunnelling GTP (GPRS Tunelling Protocol)⁶.
- ✓ Recibir paquetes de datos de la red troncal GPRS, eliminar el túnel GTP y encaminar los datos de usuario hacia Intranet o Internet.
- ✓ Garantizar privacidad, seguridad para la red y el terminal GPRS. Para ello, el GGSN actúa como una puerta de acceso entre las redes externas y la red GPRS.
- ✓ Proporcionar direcciones IP a los terminales GPRS cuando se emplea direccionamiento dinámico.
- ✓ Proporcionar los servicios básicos para el acceso a ISPs.

Un GGSN puede soportar diferentes tipos de interfaz física, entre los que son:

- ✓ **V.35**⁷: se usa en conexiones WAN en el caso de que el SGSN soporta hasta 2Mbps,
- ✓ **100BaseTX**⁸: soporta configuraciones en SGSN(s) y GGSN(s) están en el mismo lugar y conectados a una red local. La capacidad de la interfaz es de 100 Mbit/s.
- ✓ **STM-1**⁹: soporta tanto configuraciones WAN como LAN. Esta interfaz permite 155 Mbit/s.

⁵ La función es recoger los CDRs generados por los SGSN y GGSN, para procesarlo antes de enviar al sistema de tarificación.

⁶ Es un protocolo túnel que permite a los usuarios finales de una red GSM o UMTS transportarse de un lugar a otro lugar sin desconectarse a Internet.

⁷ Es una interfaz serie de alta velocidad.

⁸ Interfaz Fast Ethernet con una velocidad de 100Mb/s, que utilizan dos pares de hilos Cat. 5 o mejores.

⁹ Módulo de Transporte Síncrono. Unidad de transmisión básica de la Jerarquía Digital Síncrona (SDH), correspondiente al primer nivel básico.

d.2.2 PROTOCOLO GPRS

El protocolo GPRS es un protocolo de nivel tres, transparente para todas las entidades de red comprendidas entre el terminal móvil MT y el nodo GSN al que el móvil está, lógicamente conectado; las entidades entre las que se establece una conexión a este nivel están, de hecho, localizadas en el terminal móvil MT y en el nodo GSN. Este protocolo soporta tanto el intercambio de informaciones de control como de paquetes PDP-PDU (Packet Data Protocol - Protocol Data Unit) entre el móvil y el nodo al que éste se encuentra conectado (los PDP-PDU son, encapsulados en las tramas GPRS) (11).

d.2.2.1 PILA DE PROTOCOLOS DE SM

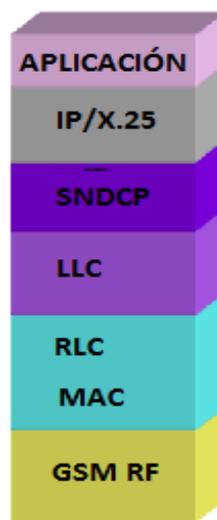


Figura 4: Pila de protocolos de una estación móvil. Fuente: (12).

La descripción de la figura, se presenta a continuación:

APLICACIÓN.- Es la interfaz que utiliza el usuario para realizar el envío de datos a través de la red.

IP/X.25.- Es la interfaz entre el de la estación móvil de datos y el equipo de terminación del circuito de datos; para terminales móviles que funcionan en el modo paquete y están conectados a redes públicas que establece mecanismos de direccionamiento entre usuarios, negociación de características de comunicación y técnicas de recuperación de errores.

SNDCP.- Este protocolo es el encargado de la segmentación, reensamblado, compresión, encriptación y multiplicación en una única conexión virtual de los mensajes de datos de usuario y de control del nivel de red (IP, X.25).

LLC.- Este protocolo proporciona un control de enlace lógico altamente fiable entre la estación móvil y el Nodo de Soporte de Servicios GPRS (SGSN), incluyendo retransmisiones, control de flujo, cifrado y corrección/detección de errores, además soporta tramas de información de longitud variable, transferencia de datos en modo confirmado y no confirmado, multiplexa datos del SGSN utilizando un mismo recurso radio, también permite y habilita los parámetros de QoS (calidad de servicio) a usar en cada conexión.

RLC.- Protocolo responsable de la transmisión de datos en el espectro radioeléctrico y de los procedimientos de control de errores (BEC) mediante la retransmisión selectiva de bloques.

MAC.- Protocolo encargado de controlar el acceso a los procedimientos de señalización (petición y asignación) de los canales radio compartidos por las MS y gestión del mapeo de tramas LLC sobre los canales físicos de la trama GSM.

GSM RF.- Hace referencia a los aspectos físicos de la red que se utiliza para la transmisión de los datos (medios de transmisión, dispositivos que conforman la parte física de la red, entre otros.).

d.2.2.2 PROTOCOLO HTTP

Se define como un Protocolo de Transferencia de Hipertexto, es uno de los más populares por su uso en cada transacción de la World Wide Web, también se dice que es un protocolo sin estado, es decir que no guarda ninguna información sobre conexiones anteriores, además hace uso del puerto 80. La sintaxis de petición del protocolo HTTP es la siguiente:

```
``http :? `` // `` dirección [``: `` puerto] [path]
```

Donde

dirección es el Sistema de Nombres de Dominio (DNS) o una dirección IP, ***puerto*** (por defecto 80), y

path (por defecto “/”) indica el recurso al que se accede.

Este protocolo tiene diferentes versiones de HTTP que pueden ser la 1.0, la 0.9 y la 1.1 en la actualidad la más usada es la HTTP/1.1 y sus métodos de peticiones son: Get, Head, Post, Put, entre otros. Un mensajes HTTP consisten en hacer una petición desde el cliente al servidor y por ende recibirá una respuesta del servidor al cliente, estas peticiones se las clasifica de dos formas y son simples o completas. “En el caso de que el servidor tenga implementado un método pero no está permitido para el recurso que se pide, entonces se devuelve un código de estado 405 (método no permitido), si lo que ocurre es que no tiene implementado el método, entonces devuelve un código 501 (no implementado). Los únicos métodos que deben soportar los servidores de forma obligatoria son los métodos GET y HEAD” (13).

d.2.2.3 PROTOCOLO GTP

Es un protocolo de túnel, que generalmente se encuentra en el núcleo de una red inalámbrica celular que está compuesto por un plano de control (GTP-C) y un plano de usuario (GTP-U). Su plano GTP-C implementa una cantidad de mensajes considerable. Los mensajes se intercambian entre diferentes elementos de la red celular y están agrupados en base a quien los intercambia, mientras que GTP-U por su parte solamente implementa dos tipos de mensajes, Echo Request que se envía a un terminal el cual responderá con un mensaje de Echo Response que contendrá los mismos datos.

d.2.2.4 TRAMA GPRS

La tecnología GPRS separa las asignaciones de recursos entre enlace ascendente como descendente. GPRS solo usará un canal cuando envíe o reciba un paquete de información, el resto del tiempo el canal estará inactivo para que pueda ser usado por otro usuario. Además define la interfaz radio basada en TDMA¹⁰ para proporcionar transmisión de paquetes sobre la interfaz de aire Um, a través de esta se conectan las MS con la red GSM, para ello se utilizan las BTS como punto de conexión con la red.

¹⁰ Es una técnica que permite la transmisión de paquetes de datos sobre la interfaz de aire, estableciendo de esta manera, nuevas formas de usar los canales de radio GSM ya existentes, logrando de esta manera un mejor aprovechamiento de medio de transmisión.

En la figura se muestra la trama TDMA subdividida en 8 intervalos de tiempo o también conocidos como time slot.

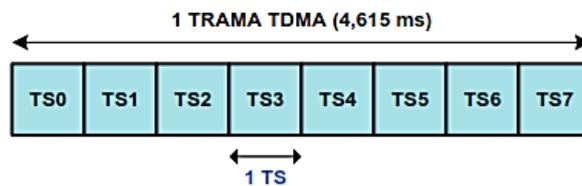


Figura 5: Estructura de trama TDMA. Fuente: (14)

En GPRS se establecen procedimientos a través de los cuales múltiples usuarios pueden compartir simultáneamente los recursos de radio y las ranuras de tiempo de la trama. GPRS incluye algunas modificaciones en la interfaz radio de GSM, las mismas que se describen a continuación:

- ✓ Tiene la capacidad de asignar de forma dinámica los canales de tráfico dejando libre canales GSM como canales de datos.
- ✓ Asigna canales bajo demanda para distribuir los recursos de radio de manera eficiente.
- ✓ La capa de acceso al medio realiza el monitoreo de la carga de los canales de datos en las celdas y de acuerdo a esto, aumentar o disminuir el número de canales de tráfico asignados para la transmisión de paquetes.
- ✓ Los canales tanto ascendente como descendente son asimétricos e independientes.
- ✓ Mecanismo para solicitar un canal aleatoriamente para transmisión de paquetes de datos.
- ✓ Sincronizar MS con la BTS tanto en frecuencia y tiempo.
- ✓ Monitoreo del nivel de potencia de otra radio bases cercanas para el *handover*.
- ✓ Minimizar interferencias.
- ✓ Protección de los datos frente a errores (codificación) e intrusos (cifrado).
- ✓ Un canal sólo se asigna cuando se transmite o recibe (en GSM se asigna de forma permanente durante toda la llamada).
- ✓ Las estaciones móviles pueden usar simultáneamente de 1 a 8 intervalos de tiempo de la trama TDMA para la transferencia de paquetes, incrementando así la capacidad de transmisión del sistema GPRS (14).

d.2.2.5 MODULACIÓN GPRS

GPRS utiliza la modulación GMSK (Modulación por desplazamiento mínimo gaussiano); es un esquema de modulación binaria simple que se engloba dentro de las modulaciones PSK, siendo una variante de MSK. En GMSK, los lóbulos laterales del espectro de una señal MSK se reducen pasando los datos modulantes a través de un filtro Gaussiano de pre-modulación, con el objetivo de reducir el ancho de banda y de esta manera mejorar su eficiencia espectral e interferencias producidas por canales adyacentes. “El filtro gaussiano aplana la trayectoria de fase de la señal MSK y por lo tanto, estabiliza las variaciones de la frecuencia instantánea a través del tiempo. Esto tiene el efecto de reducir considerablemente los niveles de los lóbulos laterales en el espectro transmitido. En la práctica, GMSK es muy atractiva por su excelente eficiencia de potencia y espectral. El filtro de pre-modulación introduce interferencia intersimbólica ISI ("Inter-Symbol Interference") en la señal transmitida, pero esta degradación no es grave si el parámetro BT del filtro es mayor de 0.3" (11).

d.2.2.6 CODIFICACIÓN GPRS

Para la codificación GPRS existen cuatro tipos diferentes cada uno con características propias, a continuación la figura para la codificación GPRS.



Figura 6: Pasos para codificación. Fuente: (El Autor)

En los dos primeros bloques o etapas luego de la carga útil se añade información:

BCS.- Secuencia de Chequeo de bloque.

USF.- Permite el multiplexado uplink. Consta de 3 bits por lo que tiene 8 valores diferentes

Una vez ya obtenido el código se procede hacer un diezmado, para hacer esto se quita bits de la carga útil de manera no arbitraria. Entonces los cuatro tipos de codificación son:

Tabla 2: Tipos de codificación de datos GPRS. Fuente: (11)

Tipo	Tasa Código	Carga Útil	BCS	USF	Cola	Bits codif.	Bits diezm.	Tasa de datos (kbps)
CS-1	1/2	181	40	3	4	456	0	9,05
CS-2	≈2/3	268	16	6	4	588	132	13,4
CS-3	≈3/4	312	16	6	4	676	220	15,6
CS-4	1	428	16	12	0	456	0	21,4

d.2.3 CLASE DE TERMINALES MÓVILES

Los terminales de las redes celulares se los clasifican de acuerdo a los servicios que brindan tanto GPRS (datos) como GSM (voz), siendo uno a la vez o pudiendo utilizarlos simultáneamente. Al utilizar los dos servicios simultáneamente es probable que exista una degradación del rendimiento en cuanto al throughput de la llamada tecnología GPRS. En base a este propósito se los clasifica en:

d.2.3.1 CLASE A

Este terminal tiene la capacidad de soportar tanto GSM (voz) como GPRS (datos) simultáneamente, para lo que utilizará un timeslot para GSM y más de un timeslot para GPRS, con el objetivo de no degradar los servicios.

d.2.3.2 CLASE B

Tiene la posibilidad de transmitir, recibir servicio de datos o voz pero no de manera simultánea; ya que si el terminal recibe una llamada cuando se encuentra en modo de paquetes, el usuario será el que determine dar paso a la llamada o continuar con la conexión anterior.

d.2.3.3 CLASE C

El terminal tiene dos opciones ya sea transmisión de datos o únicamente para voz, manteniéndose activos solo en uno de los dos tipos de comunicación, la elección del servicio se lo deberá hacer manualmente.

d.2.4 COMANDOS AT

Los comandos AT fueron desarrollados por Dennis Hayes por los años de 1977 como un conjunto de instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un dispositivo terminal (Modems).

Se los consideró como una interfaz de comunicación con un terminal que acepte este tipo de códigos, para poder configurarlo y proporcionarle instrucciones, para que realicen actividades como: marcar un número de teléfono, enviar un mensaje, enviar datos, entre otros.

En posteriores adelantos y expansión sobre estos comandos fueron realizadas por las compañías Microcomm y US Robotics, con un solo objetivo de universalizarlos y puedan ser utilizados para trabajos en áreas de telemando, telemetría, etc. Los comandos AT se denominan así por la abreviatura de attention que hace referencia al Modem a poner atención a la solicitud de comando.

Aunque la finalidad principal de los comandos AT es la comunicación con modems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un conjunto de comandos AT específicos que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal (14).

d.2.4.1 SINTAXIS DE LOS COMANDOS AT

En estos comandos se escribe primeramente el prefijo “AT” ya que se puede escribir más instrucciones en la línea de comandos y solamente una vez se escribirá el prefijo, también están compuestos por cadenas de caracteres ACSII dentro de la línea de comandos.

Estos comandos AT tienen una estructura establecida para la ejecución de los mismos que conoceremos a continuación:

AT+	CGATT=1	<CR>
Prefijo	Comando	Sufijo

- ✓ El **prefijo** de los comandos AT debe ser la cadena de caracteres “AT”, con el signo“+”.
- ✓ El **comando** es la solicitud que se procesa en el terminal GSM/GPRS y puede escribirse con letras minúsculas o mayúsculas no influye en la solicitud de respuesta.
- ✓ El **sufijo** de los comandos AT debe ser retorno de carro <CR>, es el equivalente a ENTER.

En los comandos existen los siguientes modelos de solicitudes:

AT+ = **x** Configura un parámetro con valor x.

AT+ ? Pide información al modem por ejemplo puede ser calidad de señal, etc.

AT+ = ? Se obtiene todo el rango de valores posibles que se pueden configurar.

d.2.4.2 COMANDOS

A continuación describiremos los comandos AT generalmente usados:

- ✓ **Comandos de información.**

AT+CGMI: Identificación del fabricante.

AT+CGSN: Obtener número de serie.

AT+CIMI: Obtener el IMSI (*Identificación de la Estación Móvil Internacional*).

AT+CPAS: Leer estado del modem.

- ✓ **Comandos del servicio de red**

AT+CSQ: Obtener calidad de la señal.

AT+COPS: Selección de un operador.

AT+CREG: Registrarse en una red.

AT+WOPN: Leer nombre del operador.

- ✓ **Comandos de seguridad**

AT+CPIN: Introducir el PIN.

AT+CPINC: Obtener el número de reintentos que quedan.

AT+CPWD: Cambiar password.

✓ **Comandos para SMS**

AT+CPMS: Seleccionar lugar de almacenamiento de los SMS.

AT+CMGF: Seleccionar formato de los mensajes SMS.

AT+CMGR: Leer un mensaje SMS almacenado.

AT+CMGL: Listar los mensajes almacenados.

AT+CMGS: Enviar mensaje SMS.

AT+CMGW: Almacenar mensaje en memoria.

AT+CSCA: Establecer el Centro de mensajes a usar.

✓ **Comandos para envío de datos por GPRS**

AT+SAPBR: Ajuste de portador para las aplicaciones basadas en IP.

AT+HTTPINIT: Inicialización de servicio HTTP.

AT+HTTPTERM: Terminación de servicio HTTP.

AT+HTTPPARA: Conjunto de valores de parámetros HTTP.

AT+HTTPDATA: Entrada de datos HTTP.

AT+HTTPACTION: Método de acción HTTP

AT+HTTPREAD: Lee las respuestas del servidor HTTP

AT+HTTPSCONT: Graba el contexto de aplicaciones HTTP.

d.3 GITHUB

d.3.1 CONTROL DE VERSIONES

El control de versiones es un sistema que permite alojamiento de archivos en la nube de manera gratuita o con un costo determinado de acuerdo a la prioridad y seguridad de los mismos, además registra los cambios realizados sobre estos a lo largo del tiempo, con la opción de poder recuperar versiones específicas. También permite obtener un diseño web en caso de requerirlo; además tiene la facilidad de actualizar el diseño desarrollado con cada una de la versión cargada, y también permite revertir sus archivos o el proyecto entero a un estado anterior.

d.3.1.1 HISTORIA DE GIT

Linux nace como un proyecto de software de código abierto con un alcance bastante grande, durante el desarrollo y mantenimiento de este (1991-2002), se desarrollaron cambios en el software que se pasaron en forma de parches y archivos. A partir del año

2002, se empezó a usar un Sistema de control de Versiones Distribuidos (DVCS) denominado BitKeeper basado en el núcleo de Linux.

Tiempo más tarde (2005) se desvaneció la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía que desarrollaba BitKeeper lo que produjo que esta herramienta deje de ofertarse gratuitamente, fue entonces cuando la comunidad de Linux sintió la necesidad de desarrollar su propia herramienta basada en algunas experiencias obtenidas en base al uso de BitKeeper. Para el desarrollo de esta nueva herramienta denominada GitHub comandada por Linus Torvalds, se plantearon algunos objetivos con el afán de ofrecer un Sistema de Control de versiones confiable, entre estos tenemos los siguientes:

- ✓ Mayor velocidad que los DVCS.
- ✓ Diseño sencillo.
- ✓ Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas).
- ✓ Completamente distribuido.
- ✓ Capaz de manejar grandes proyectos de manera eficiente.

Git en la actualidad ha evolucionado y madurado tremendamente siendo uno de los más grandes sistemas de control de versiones, fundamentándose en sus objetivos con el afán de servir a la comunidad con beneficios como: uso sencillo, altas velocidades, eficiente a la hora de alojar grandes proyectos, ofreciendo un increíble sistema de ramificación para desarrollo no lineal.

d.3.2 FUNDAMENTOS DE GIT

Git se fundamenta en como almacena y modela la información de forma muy diferente a otros sistemas como: CVS, Subversion, Perforce, Bazaar, etc. A pesar de que su interfaz sea bastante similar; conceptualmente la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos o como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo, en la figura se muestra la manera que estos almacenan la información.

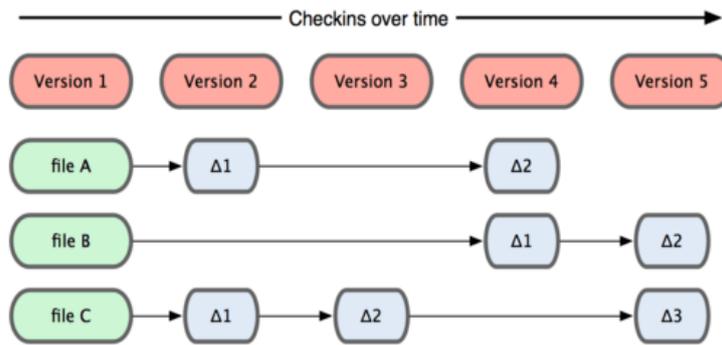


Figura 7: Sistemas que almacenan datos como cambios de cada archivo. Fuente: (15).

Git no modela ni almacena sus datos de este modo. Lo hace como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que se confirma un cambio, o guardas el estado de tu proyecto. Git básicamente hace una foto del aspecto de todos los archivos en ese momento y guarda una referencia del proyecto, si los archivos no se han modificado Git no almacena el archivo de nuevo, sólo hace un enlace al archivo anterior idéntico que ya tiene almacenado. Git modela sus datos como se muestra en la figura 7 (15).

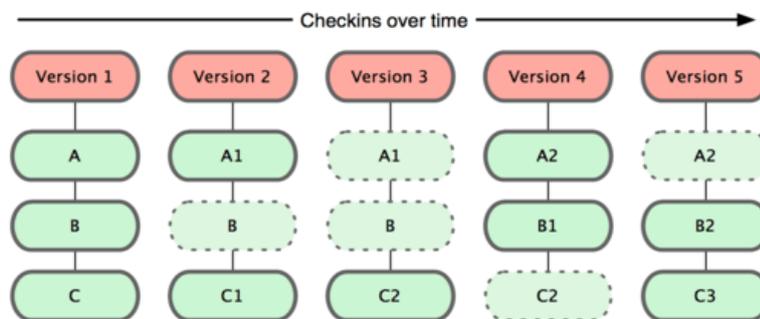


Figura 8: Git almacena la información como instantáneas. Fuente: (15).

Git contiene tres estados principales en los que se pueden encontrar tus archivos, estos son: confirmado (committed), modificado (modified), y preparado (staged).

Confirmado.- Los datos están almacenados de manera segura en la base de datos local.

Modificado.- Se ha modificado el archivo pero todavía no lo has confirmado en la base de datos.

Preparado.- Se ha marcado un archivo modificado en su versión actual para que vaya en la próxima confirmación.

Por tanto tenemos tres secciones importantes en el proceso de operación de un proyecto en Git que son: el directorio de Git, el directorio de trabajo y el área de preparación.

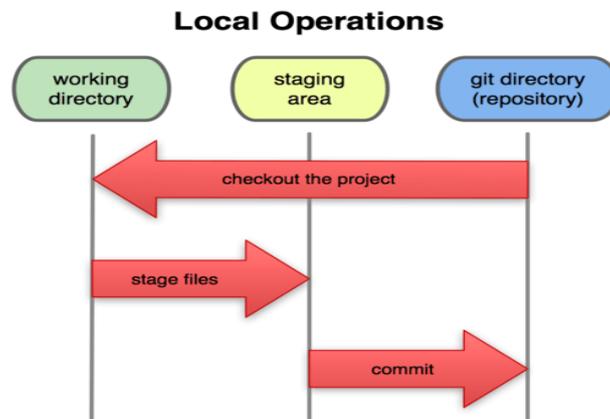


Figura 9: Directorio de trabajo, área de preparación y directorio de Git. Fuente: (15).

El **directorio de Git** es donde se almacena los metadatos y la base de datos de objetos para un proyecto, también se la puede clonar y tener un repositorio en tu ordenador.

El **directorio de trabajo** es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en un disco para que los puedas usar o modificar.

El **área de preparación** es un sencillo archivo, generalmente contenido en el directorio de Git, que almacena información acerca de lo que va a ir en tu próxima confirmación.

d.3.2.1 INSTALACIÓN DE GIT

La instalación de Git se la puede realizar en diferentes sistemas operativos tanto para Windows, MAC y en el mismo Linux en estos dos últimos se los puede instalar en base a código a través de la consola o terminal, para ello necesitas instalar algunas librerías en el caso de Linux como: curl, zlib, openssl, expat y libiconv.

Para instalar **Git en Linux** a través de un instalador binario (consola), en general puedes hacerlo a través de la herramienta básica de gestión de paquetes que trae tu distribución.

- ✓ Si utiliza Fedora, puedes usar yum:

\$ yum install git-core

- ✓ si utiliza una distribución basada en Debian como Ubuntu, prueba con apt-get:

\$ apt-get install git

Para instalar **Git en Mac** existen las siguientes maneras.

- ✓ Una alternativa es instalar Git a través de MacPorts con el siguiente comando.

\$ sudo port install git-core +svn +doc +bash_completion +gitweb

- ✓ Otra alternativa es a través de Homebrew:

\$ brew install git

- ✓ La más sencilla es usar el instalador gráfico de Git.

Instalar **Git en Windows** es bastante fácil y útil. Existen dos procesos de instalación, uno de ello es descargar el instalador desde la página de GitHub, y ejecútalo, mientras que el otro se lo puede hacer instalando la consola Git Bash en Windows, la misma que permitirá subir archivos a GitHub a través de línea de comandos, para lo cual debes sincronizar estas dos herramientas, siendo esta una excelente opción de trabajo con la que se desarrolló el presente proyecto.

d.3.2.2 CONFIGURACIÓN DE GIT

Git trae una herramienta llamada **git config** que permite obtener y establecer variables de configuración, que controlan el aspecto y funcionamiento de Git.

Las configuraciones básicas que se debe hacer son las siguientes:

Establecer tu nombre de usuario y dirección de correo electrónico.

\$ git config --global user.name "nombre"

\$ git config --global user.email @example.com

d.3.2.3 GENERAR UNA CLAVE PÚBLICA SSH

Para trabajar con Git es necesario poseer una autenticación a través de claves públicas SSH, estas se las puede generar desde la consola y se almacenan dentro de la carpeta del usuario SSH en el ordenador con sistema operativo Windows, utilizando la siguiente extensión `~/.ssh`. Para generar una clave debes ingresar la siguiente línea de comando:

\$ ssh-keygen -t rsa -C "email"

Acto seguido se debe ingresar dicha clave en la cuenta de Git y se comprueba si se encuentra activa con la siguiente sintaxis:

```
$ ssh -T git@github.com
```

Si deseamos conocer si ya tenemos una clave SSH ingresamos al directorio antes mencionado y listamos los archivos de la siguiente manera:

```
$ cd ~/.ssh
```

```
$ ls
```

En caso de observar dos archivos “idrsa o iddsa” y otro “.pub” en este último te indica que esa es tu clave privada SSH; en caso de no tener estos archivos deberás realizar el proceso anteriormente descrito.

d.3.2.4 CREAR UN REPOSITORIO GIT

Puedes obtener un repositorio de un proyecto en Git de dos maneras.

La primera se puede inicializar un repositorio en un directorio existente únicamente yendo al directorio donde se encuentra y tipiendo el comando:

```
$ git init
```

Crea un nuevo subdirectorio llamado .git que contiene todos los archivos necesarios del repositorio, acto seguido se puede añadir el nuevo archivo con un comentario de la siguiente manera:

```
$ git add README // crea un archivo llamado README.
```

```
$ git commit -m “versión inicial del proyecto” // crea un comentario.
```

Esto hace que tengamos un repositorio Git con archivos bajo seguimiento, y una confirmación inicial.

La segunda es en base a una copia de un repositorio Git existente, para lo que se necesita tipiar el comando **git clone** y de esta manera se obtienes una copia casi completa de un repositorio cualquiera. El comando para obtener un repositorio es de la siguiente manera:

```
$ git clone git://github.com/[nombre del repositorio]/[nombre del directorio].git
```

Con este comando se descarga toda la información de un repositorio, realizando en si una copia de la última versión de un trabajo.

d.3.2.5 GUARDANDO CAMBIOS EN EL REPOSITORIO

Una vez que se tenga un repositorio Git completo ya sea inicializado o clonado y luego de haber procedido a realizar cambios se debe confirmar las instantáneas de esos cambios a tu repositorio. Es importante tener en cuenta que el repositorio puede estar en uno de estos dos estados: **bajo seguimiento** (tracked), quiere decir que se encontraban en la última instantánea y pueden haber sido modificados o no, o **sin seguimiento** (untracked), estos son cualquier archivo del directorio que no estuvo en la última instantánea.

Cabe mencionar que la primera vez que se clona un repositorio, todos los archivos estarán bajo seguimiento y sin modificaciones, ya que se los acaba de copiar el repositorio y no has modificado ningún archivo.

A medida que editas archivos Git automáticamente los lee como modificados, porque se ha procedido a realizar algún cambio, acto seguido preparar los archivos modificados para confirmar todos los cambios hechos, esto se torna un ciclo repetitivo. Este proceso se ilustra en la siguiente figura.

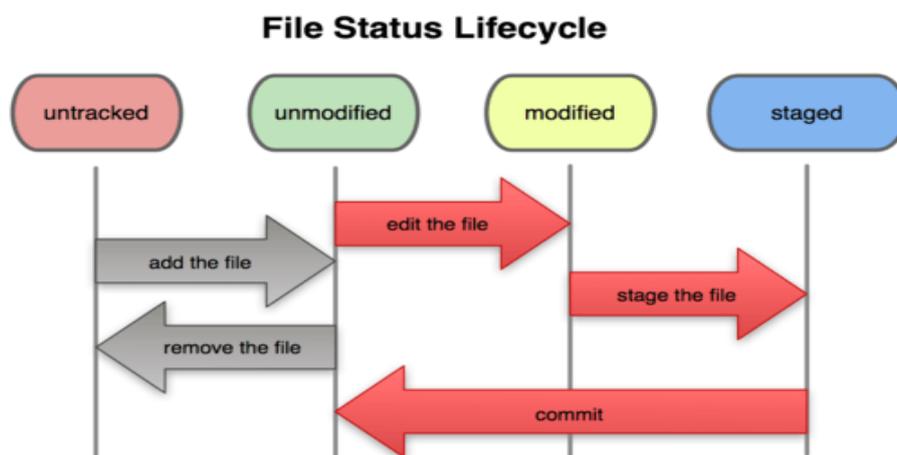


Figura 10: El ciclo de vida del estado de tus archivos. Fuente: (15).

Es importante comprobar el estado de los archivos de tu repositorio, con el comando **git status** informará si tienes archivos bajo seguimiento o si se ha realizado modificaciones en ellos.

d.3.2.6 REPOSITORIOS REMOTOS

Los repositorios remotos son versiones de un proyecto que se encuentra alojado en internet o en algún punto de la red (15). Además al crear repositorios remotos sirve para

colaborar con otros proyectos, al hacer esto significa saber cómo añadirlos para que se convierta en una versión nueva del proyecto, también debemos saber cómo eliminar aquellos que ya no son válidos y gestionar ramas remotas e indicar si están bajo seguimiento o no.

d.3.2.6.1 CREAR REPOSITORIOS REMOTOS

Para clonar un repositorio debemos hacerlo con el comodin **clone** antes de esto es necesario conocer si se tiene o no algún repositorio remoto para lo que escribiremos la siguiente sintaxis:

```
$ git remote -v // indica los repositorios remotos.
```

```
$ git clone git://github.com/[nombre del usuario]/[url].git // clona un repositorio.
```

Una vez clonado el repositorio, se puede trabajar en el generando cambios, ahora bien para añadir un nuevo repositorio remoto, se debe asignar un nombre con el que se pueda referenciarlo fácilmente, para añadirlo se ejecuta el comando **git remote add [nombre] [url]**:

```
$ git remote add pb git://github.com/[nombre del usuario]/[url].git
```

Siempre que se agregue un repositorio remoto es necesario saber si dentro de la lista de estos; el comando **git remove -v**. Ahora bien solo en caso de querer recuperar un repositorio remoto anterior deberás ejecutar el comando:

```
$ git fetch [nombre-remoto]
```

Es importante tener en cuenta que el comando fetch sólo recupera la información y la pone en tu repositorio local no la une automáticamente con tu trabajo ni modifica aquello en lo que estás trabajando. Tendrás que unir ambos manualmente a posteriori. Si has configurado una rama para seguir otra rama remota puedes usar el comando **git pull** para recuperar y unir automáticamente la rama remota con tu rama actual. Esto puede resultar un flujo de trabajo más sencillo y más cómodo; y por defecto, el comando **git clone** automáticamente configura tu rama local maestra para que siga la rama remota maestra del servidor. Al ejecutar **git pull**, por lo general se recupera la información del servidor del que clonaste, y automáticamente se intenta unir con el código con el que estás trabajando actualmente (15).

d.3.2.6.2 SUBIR EL REPOSITORIO REMOTO

Para compartir un proyecto se tiene que subir o enviarlo a un repositorio remoto, para ello se hace con el comando: **git push [nombre-remoto][nombre-rama]**. En caso de requerir enviar a la rama master, se ejecutará lo siguiente:

\$ git push origin master

d.3.2.6.3 GESTIONAR UN REPOSITORIO REMOTO

Para gestionar la información de cada repositorio remoto se debe utilizar la siguiente sintaxis: **git remote show [nombre]**. Por ejemplo puedes utilizar el repositorio origen de la siguiente manera:

\$ git remote show origin

Esto lista la URL del repositorio remoto, así como la información sobre las ramas bajo seguimiento. Este comando te recuerda que si estás en la rama maestra y ejecutas **git pull**, automáticamente unirá los cambios a la rama maestra después de haber recuperado todas las referencias remotas.

d.3.2.6.4 RENOMBRAR Y ELIMINAR REPOSITORIOS REMOTOS

Para renombrar una referencia a un repositorio remoto en versiones recientes de Git se debe ejecutar la siguiente línea de comando **git remote rename**. Por ejemplo, si quieres renombrar [nuevo nombre] a [nombre actual], puedes hacerlo de la siguiente manera:

\$ git remote rename [nuevo nombre] [nombre actual]

Cabe recalcar que esto también generara cambios de nombre en las ramas remotas, por lo que quedará de la siguiente manera: [nombre del archivo]/master y posteriormente cambiará al [nombre del archivo actual]/master.

Ahora bien en caso de querer eliminar una referencia por cualquiera que fuese el caso (versión antigua, no estas contribuyendo más a ese repositorio, tiene algún error, etc.) se escribirá la siguiente sintaxis:

\$ git remote rm [nombre del archivo]

d.3.2.7 HISTORIAL DE LAS CONFIRMACIONES

El historial de confirmaciones nos mostrará cuantos commit se produjeron dentro de un repositorio en algunas ocasiones es muy necesario mirar la cantidad de estos para corregir un error para lo cual se hace uso del comando **git log**.

Este comando proporciona gran cantidad de comodines para mostrarte exactamente lo que buscas, dentro de las más utilizadas tenemos el comodin `-p`, que muestra las diferencias introducidas en cada confirmación y el comodin `-2`, que muestra únicamente los dos últimos seguimientos. También se puede ver este historial de manera gráfica a través del comando **Gitk** que básicamente es un git log pero de manera gráfica.

d.3.3 RAMIFICACIONES EN GIT

Todo CVS moderno diseña un mecanismo que le ayude con el control de sus distintas ramificaciones; al hablar de estas debemos tener en cuenta que se tendrá una rama principal (master). Por lo que es muy importante aprender el modelo de Git para realizar las ramificaciones, si tomamos en cuenta lo descrito anteriormente Git no almacena de forma incremental, sino como una serie de instantáneas (copias puntuales de los archivos completos, tal y como se encuentran en ese momento).

Una rama Git es simplemente un apuntador móvil, apuntando a una de esas confirmaciones. La rama por defecto de Git es la rama master. Con la primera confirmación de cambios que realicemos, se creará esta rama principal y apuntará a dicha confirmación. En cada confirmación de cambios que realicemos, el número de ramas irá avanzando automáticamente y la rama master apuntará siempre a la última confirmación realizada (15).

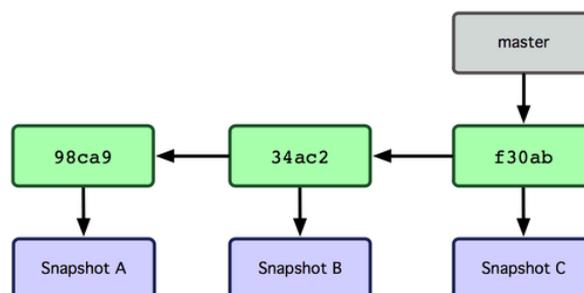


Figura 11: Apuntadores en el registro de confirmaciones de una rama. Fuente: (15).

d.3.3.1 PROCEDIMIENTOS BÁSICOS DE RAMIFICACIÓN

La creación de ramas es principalmente para trabajar en mejoras de un proyecto, este puede ser un diseño web o simplemente complemento de datos que se esté alojando en Git. Para crear una nueva rama y saltar a ella, en un solo paso, puedes utilizar el comando **git checkout**, con la opción **-b** se crea una nueva rama.

\$ git checkout -b (nombre de la nueva rama).

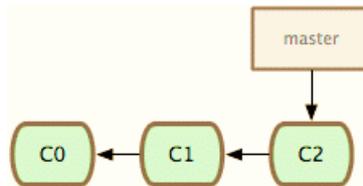


Figura 12: Un registro de confirmaciones simple y corto. Fuente: (15).

Es recomendable que el estado de trabajo de la rama master u otra rama en la que se encuentre deberá estar limpio y despejado antes de saltar entre ramas. El comando para saltar a otra rama o saltar a la master se lo escribe de la siguiente manera:

\$ git checkout (nombre de la rama a saltar)

Para realizar la limpieza del estado de trabajo se hace uso de algunos comodines como: **stash**, **commit** y **amend**, que serán favorables para poder saltar a otra rama sin inconveniente alguno.

d.3.3.2 PROCEDIMIENTOS BÁSICOS DE FUSIÓN

Una vez creada la rama en algunos casos puede ser necesario fusionarle con la rama master para que la nueva rama se incorpore y así mostrar una nueva actualización en el repositorio. Para realizar este proceso se hace uso del comando **git merge**, cabe recalcar que para realizar esta fusión es necesario estar en la rama principal o master, entonces la sintaxis será la siguiente:

\$ git checkout master // salto a la rama master.

\$ git merge (nombre de la rama a incorporar) // fusión de la rama.

Una vez realizado con éxito el proceso no hay la necesidad de contar con esta rama por lo que se puede borrar escribiendo la siguiente línea de comando:

\$ git branch -d (nombre de la rama a incorporar)

d.3.3.3 GESTIÓN DE RAMIFICACIONES

Una vez realizado el proceso anterior veremos algunas herramientas de gestión bastante útiles como por ejemplo: **branch**, que con comodines adicionales puedes crear y borrar las ramas, pero al utilizarle sin argumentos nos listará las ramas existentes dentro de un proyecto, para ello se emplea la siguiente sintaxis:

\$ git branch

Dentro del resultado que nos presente este comando debemos tomar en cuenta que cuando muestre un asterisco al inicio de la rama denotará la rama que se encuentra activa en ese momento, mientras que para gestionar la última confirmación de cambios producidos en la rama se escribirá la siguiente sintaxis:

\$ git branch -v

De esta manera podemos gestionar nuestras ramas y saber cómo se ha ido actualizando cada una de ellas dentro del proyecto.

e MATERIALES Y MÉTODOS

e.1 MATERIALES

El desarrollo de las nuevas tecnologías hoy en día hace que se disponga de una gran cantidad de hardware prediseñado listo para ser utilizado, facilitando el desarrollo o implementación de proyectos, inculcados al fortalecimiento del aprendizaje, en este apartado se dará a conocer primeramente el diagrama de bloques de una estación meteorológica y cada uno de los materiales que la componen como por ejemplo: la tarjeta Arduino, sensores (temperatura, humedad, precipitación, velocidad y dirección del viento) y una tarjeta SIM900 GSM/GPRS que será utilizada para la transmisión de los datos obtenidos lo que es de gran importancia en el desarrollo de este proyecto.

e.1.1 DIAGRAMA DE BLOQUES DE PROTOTIPO IMPLEMENTADO

La estación meteorológica está constituida por cuatro bloques fundamentales, como son:

- ✓ Adquisición de datos.
- ✓ Procesamiento de datos.
- ✓ Transmisión.
- ✓ Recepción.

Como se observa en la figura 13, en donde se muestra el diagrama del proyecto, el objetivo fundamental del mismo es medir algunas variables meteorológicas como: temperatura, humedad relativa al aire, velocidad y dirección del viento y precipitación; a través de sensores electrónicos adecuados, tanto digitales como analógicos, una vez adquiridos estos datos pasan a ser procesados, en un hardware denominado Arduino UNO, el que nos permite procesarlos y almacenarlos haciendo la función de un datalogger, como siguiente paso está la transmisión de los datos que se recogieron a través de un medio inalámbrico utilizando la red celular, por el cual se transmiten los datos hasta un receptor en donde se presentarán en una adecuada aplicación que se desarrollara con ayuda de un servidor prediseñado de acceso gratuito, es de vital importancia tener conectividad a internet en el dispositivo empleado, para visualizar de manera gráfica el estado de las variables climáticas. También cabe mencionar que las áreas cubiertas por las redes inalámbricas celulares son extensas lo que hace favorable la utilización de esta red para la transmisión

de los datos adquiridos en el prototipo de estación meteorológica.

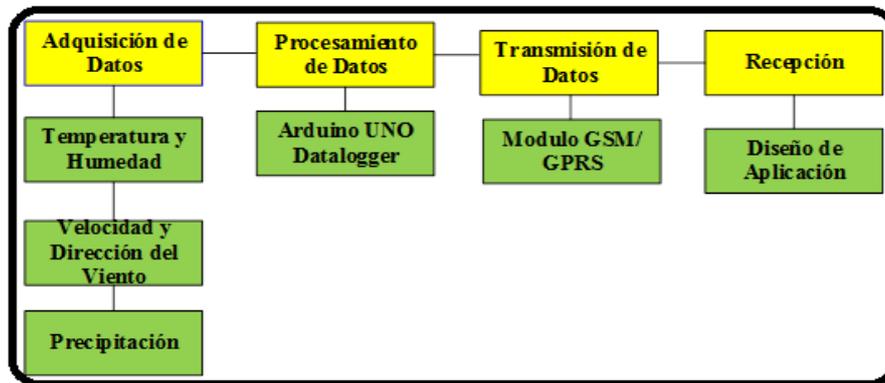


Figura 13: Diagrama de bloques del prototipo de estación meteorológica. Fuente: (El Autor).

e.1.2 PLATAFORMA ARDUINO

Arduino nace como un proyecto educativo por el año 2005 sin pensar que algunos años más tarde se convertiría en líder del mundo, su nombre proviene del nombre del Bar di Re Arduino donde Massimo Banzi pasaba algunas horas. Banzi aclara que nunca se pensó en una idea de negocio aunque hoy en día Arduino tiene la tarea de subsistir comercialmente y en continuo crecimiento tecnológico.

En la actualidad el desarrollo de prototipos se basa en software y hardware flexibles y fáciles de usar, Arduino ha sido uno de los mayores promotores en este desarrollo, ya que presenta una plataforma amigable para el progreso de nuevos proyectos. La creación de éste, ayudó al desarrollo de proyectos con mayor facilidad, además con esta placa se puede trabajar tomando información a través de sus pines de una gran variedad de sensores que son utilizados para, control de luces, control de motores, estudios climatológicos, entre otros. Además Arduino es de libre distribución al que le acoge un amplio catálogo de herramientas de programación, también de software libre convirtiéndose en una de las más importantes comunidades de usuarios en la que se intercambia importante información para el continuo adelanto tecnológico. El microcontrolador de la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring¹¹) y el entorno de desarrollo Arduino (basado en

¹¹ Marco de programación de código abierto para microcontroladores.

Processing¹²). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador (16).

Las principales ventajas para ser la parte central del diseño es la factibilidad que ofrece con: captura de datos, acondicionamiento de señal, transmisión de datos y disponibilidad de sensores compatibles con la plataforma.

e.1.3 ARDUINO UNO (ATMEGA328)

Arduino Uno es una placa electrónica basada en el microprocesador Atmega328, la misma que cuenta con 14 pines digitales de entrada/salida digital (6 pines les puede usar como salida PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB y alimentarla con una fuente de voltaje de AC a DC o también una batería que suministre voltaje y corriente adecuados.

e.1.3.1 VOLTAJE DE ALIMENTACIÓN

La alimentación se puede hacer a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente, por USB o de un adaptador de corriente alterna a corriente directa (AC-DC).

La tarjeta puede funcionar dentro de un rango de voltaje de 7 a 12 V, si sobrepasa este voltaje puede producirse un calentamiento de la misma a tal punto que se puede dañar, por lo que se aconseja tener en cuenta no exceder este rango.

Si nos fijamos en la figura 14 que se presentará más adelante algunos pines de alimentación que describiremos a continuación:

- ✓ **V_{in}**: Se puede suministrar tensión a través de este pin, de 5 V, cuando es alimentado a través de una fuente de alimentación externa, y no por conexión USB.
- ✓ **5V**: Este pin tiene una salida de 5Vcc estable, que brinda el regulador de la tarjeta.

¹² Es un lenguaje de programación y entorno de desarrollo integrado al código abierto basado en Java.

- ✓ **3.3V:** Tiene un suministro de 3,3Vcc generados por el regulador. La corriente máxima que puede entregar es de 50 mA.
- ✓ **GND:** Tiene 2 pines para tierra (17).

e.1.3.2 MEMORIA

El Arduino tiene una memoria ATmega328 de 32 KB, 2 KB de Memoria Estática de Acceso Aleatorio (SRAM) y 1 KB de Memoria de solo lectura programable y borrable eléctricamente (EEPROM).

e.1.3.3 ENTRADAS Y SALIDAS

Está compuesto por 14 pines digitales, que se pueden utilizar como entrada o salida, trabajan a 5V, pueden entregar o recibir un máximo de 40 mA. A continuación se especifican algunas funciones como:

- ✓ El pin 0 que es para recepción (RX) y el pin 1 para transmisión (TX), de datos en serie por hardware.
- ✓ Los pines para interrupciones externas son el 2 y 3, que pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor (17).
- ✓ Pines de PWM: 4, 5, 6, 9, 10, y 11 proporcionan una modulación por ancho de pulso (PWM) con 8 bits de resolución y soportan comunicación SPI.
- ✓ Los pines 7 y 8 son de transmisión y recepción por software.
- ✓ En el pin 13 se encuentra conectado un LED, cuando el pin esta en valor de alto entonces el led está encendido y cuando esta en valor de bajo, está apagado.

La placa Arduino tiene 6 entradas analógicas, que van desde la A0 a A5, cada uno de los cuales proporcionan 10 bits de resolución (es decir 1.024 valores diferentes). (17). Como característica principal se tiene que los pines A4 y A5 pin SDA y el pin SCL, se los utiliza para la comunicación TWI utilizando la librería wire¹³.

¹³ Es una librería que permite comunicación I2C y dispositivos TWI

e.1.3.4 RELÉ DE PROTECCIÓN DE SOBRECORRIENTE USB

El Arduino Uno tiene una POLYFUSE reajutable que protege los puertos USB de su ordenador. Aunque la mayoría de los ordenadores proporcionan su propia protección interna, el fusible proporciona una capa adicional de protección. Si hay más de 500 mA aplicados al puerto USB, el fusible se abre automáticamente la conexión hasta que se elimine la sobrecarga.

e.1.3.5 COMUNICACIÓN

Un Arduino se puede comunicar con un computador, con otro Arduino, con microcontroladores. El ATmega328 ofrece comunicación en serie UART (Transmisor-Receptor Asíncrono Universal) TTL (5V), que está disponible en los pines digitales 0 (RX) y 1 (TX). Para comunicación con el computador utiliza un cable USB, también es compatible con I2C y la comunicación SPI. El software de Arduino incluye una librería wire para simplificar el uso de la I2C.

e.1.3.6 PROGRAMACIÓN

La placa Arduino viene precargado con un gestor de arranque que sirve de ayuda para subir un código nuevo sin tener un programador de hardware externo. Tiene la capacidad de pasar por alto el gestor de arranque y programar el microcontrolador a través del programador serie de circuitos (ICSP).

e.1.3.7 LIBRERÍAS

Las librerías son códigos escritos en lenguaje C que permite ampliar la funcionalidad de un programa como por ejemplo da la capacidad de leer un sensor, un decodificador o cualquier dispositivo que traiga una librería incluida permitirá facilitar la programación del mismo; estas generalmente constan de dos archivos uno que será la cabecera con “sufijo .h” y el otro que contendrá el código fuente con “sufijo .cpp”.

Tabla de características:

Tabla 3: Principales características de Arduino Uno. Fuente: (3).

Microcontroladores ATmega328	
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7 - 12V
Pines de E/S Digital	14 (de los cuales 6 proporcionan PWM)
Pines de entrada analógica	6
Corriente continua para pines E/S	40 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

En la figura 14 se muestra un Arduino UNO, el mismo que fue referenciado anteriormente. Para mayor información acerca de los datos técnicos dirigirse a los anexos.

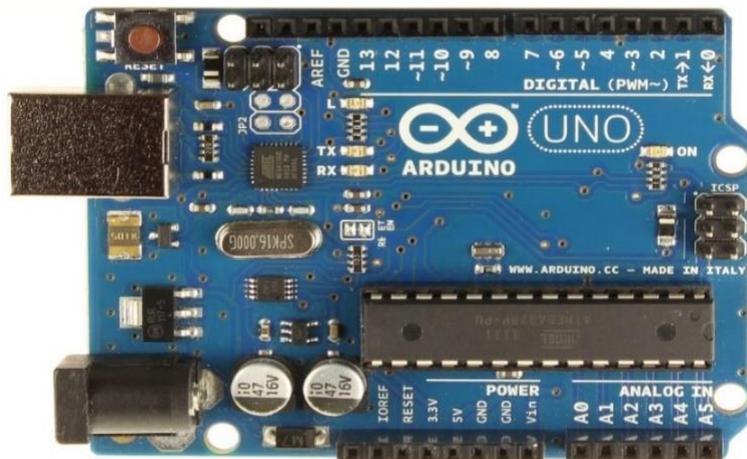


Figura 14: Arduino ATmega328. Fuente: (17).

e.1.4 MÓDULO GSM/GPRS

El Shield GSM/GPRS (Sistema Global para Comunicaciones Móviles/ Servicio General de Paquetes Vía Radio) se basa en SIM900 módulo de SIMCOM y compatible con Arduino. Este Shield, es el que proporciona la forma de comunicarse a internet mediante la red de telefonía celular GSM.

Características:

- ✓ Quad-Band 850/900/1800/1900 MHz.

- ✓ GPRS clase 10/8 multi-slot.
- ✓ GPRS Clase de estación móvil B.
- ✓ Control a través de comandos AT.
- ✓ Servicio de mensajes cortos (SMS).
- ✓ TCP Embedded / pila UDP -Le permite cargar datos a un servidor web.
- ✓ Admite Reloj en Tiempo Real (RTC).
- ✓ Puerto serie seleccionable.
- ✓ Tomas de los altavoces y auriculares.
- ✓ Bajo consumo de energía 1,5 mA (modo en reposo).
- ✓ Rango de temperatura industrial - 40°C a +85°C.

e.1.4.1 DESCRIPCIÓN DEL HARDWARE

El shield GSM/GPRS se lo dará a conocer de manera gráfica en la figura 15, a continuación describiremos algunas de sus partes:

- ✓ **Conector de energía.**- Conector de fuente de alimentación externa debe estar dentro de un rango de voltaje de 4,8 ~ 5VDC.
- ✓ **Selector de energía.**- Al cambiar de posición se selecciona la fuente que proporcionará el voltaje para el funcionamiento del módulo.
- ✓ **Conector de la antena.**- Conector para una antena externa.
- ✓ **Selector de puerto serie.**- Seleccionar software de puerto serial o puerto serie hardware esta conectado a GPRS Shield.
- ✓ **Serial Hardware.**- Los pines D0/D1 de Arduino para Rx y Tx.
- ✓ **Serial Software.**- Los pines D7/D8 de Shield SIM900 para Rx y Tx.
- ✓ **LED de estado.**- Da a conocer el estado de SIM900 si esta encendido o apagado.
- ✓ **Luz neta.**- Muestra el estado de la red de la tarjeta acerca SIM900.
- ✓ **Micrófono.**- Entrada para un dispositivo externo de audio.
- ✓ **Altavoz.**- Salida para un dispositivo de audio.
- ✓ **Botón de encendido.**- Nos permite encender o apagar el módulo SIM900.

Los pines del Arduino que se utiliza para la comunicación con el shield GPRS son:

- ✓ **D7.**- Se utiliza si se selecciona el puerto serie para recepción (Rx) del software para comunicarse con el Shield GPRS.

- ✓ **D8.-** Se utiliza si se selecciona el puerto serie para transmisión (Tx) del software para comunicarse con el Shield GPRS.
- ✓ **D9.-** Se utiliza para el control de software de la energía para un estado alto o bajo del SIM900 GSM/GPRS.

En caso de requerir mayor información técnica sobre el dispositivo dirigirse a la sección de los anexos.

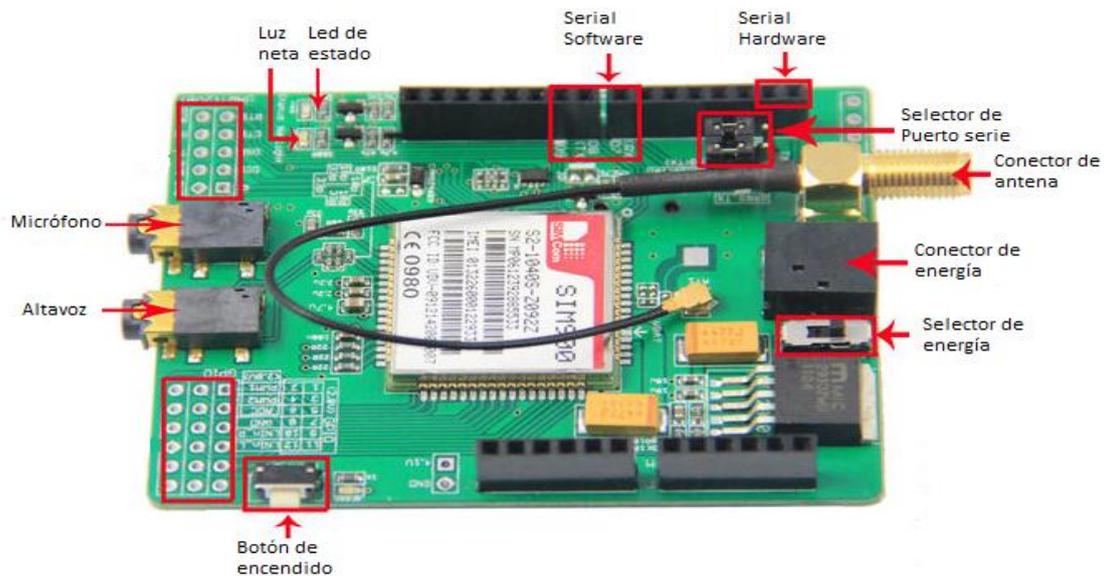


Figura 15: Descripción Shield GSM/GPRS Fuente: (18).

e.1.4.2 ENCENDIDO POR SOFTWARE

Para realizar el encendido por software se debe hacer una modificación en el hardware del módulo SIM900, que consiste en hacer la conexión en los terminales que se encuentra justo frente a los pines de TX y RX. En la figura 16 se muestra donde se debe fusionar para poder de esta manera utilizar el mando de encendido por software.



Figura 16: Conexión de hardware para modo encendido por Software. Fuente: (18)

Ahora bien para el encendido se dedica el pin 9 del microcontrolador por el que se envía una señal de inicio al módulo GSM/GPRS cambiando de estado apagado a estado de ejecución. Primeramente debemos considerar que el voltaje de alimentación del dispositivo sea de 5V; acto seguido se envía un pulso de estado bajo (0V) hace una pausa de 1000ms para dar el siguiente pulso como una orden de encendido (High) en el cual inyecta un pulso de nivel alto (5V) durante 2000ms para que se produzca el encendido (tiempo de encendido manual es de 2s) y termina la señal del pulso después de 3000ms dejando encendido el módulo GSM/GPRS.

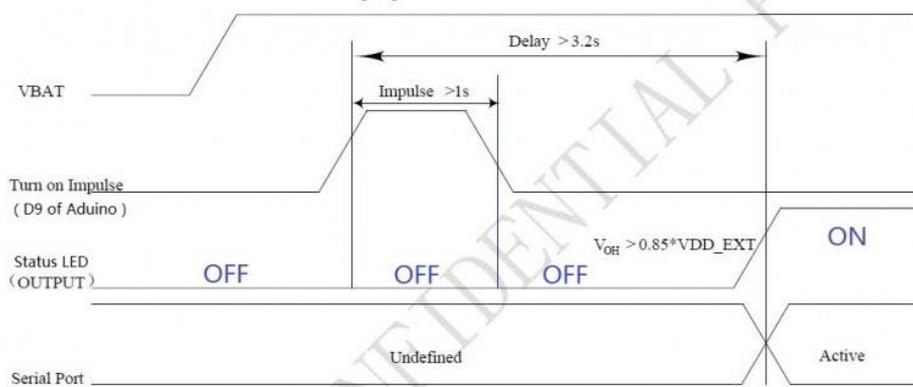


Figura 17: Sistema de encendido por software del módulo SIM900. Fuente: (19).

e.1.5 DHT22

El DHT22 es un sensor digital de temperatura y humedad, de bajo costo. Utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante o temperatura, y envía a través de una señal digital las medidas por el pin de datos. Tiene cuatro pines como se muestra en la figura 18 y su respectiva descripción se hará a continuación:



Figura 18: Sensor DHT22. Fuente: (El Autor).

Descripción de los pines del sensor DHT22 es:

Pin 1.- Ingresa el voltaje de Vcc (5V).

Pin2.- Es el pin de comunicación donde se produce el envío de los datos a la unidad de microcontrolador (ATMega328).

Pin 3.- Es un pin null que no se lo utiliza.

Pin 4.- Es el pin GND.

Para mayor información técnica acerca del sensor DHT22 diríjase a la sección de los anexos.

Tabla de características:

Tabla 4: Características técnicas de un DHT22. Fuente: (El Autor).

Descripción	Característica
Voltaje de alimentación	3.3 - 6V DC
Rango de humedad y temperatura	0 a 100%; -40°C a +80°C
Temperatura de operación	-40°C a +80°C
Precisión de humedad y temperatura	Humedad $\pm 2\%$ (Max 5% RH); temperatura $\pm 0.5^\circ\text{C}$
Resolución	humedad 0,1% de HR; 0.1°C temperatura
Estabilidad alago plazo	$\pm 0.5\%$ RH / año
Humedad histéresis	$\pm 0.3\%$ RH
Periodo de sensado	2 segundos

La figura 19 muestra la conexión de este dispositivo, como podemos observar los pines 1 y 4 son para conexión a voltaje TTL (Vcc, GND) y el pin 2 del sensor va conectado al Arduino al pin digital 4, tal como se observa en la figura, es importante tener en cuenta que el pin de datos que se lo conecta a Vcc a través de una resistencia, para disminuir la corriente que circula al pin y de esta manera mantenerlo en un nivel en alto.

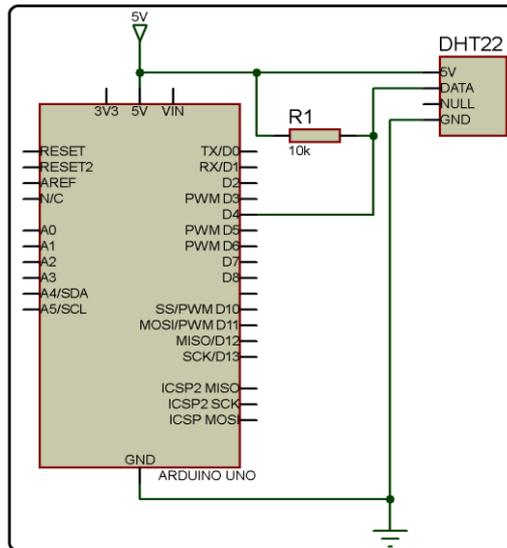


Figura 19: Esquema de conexión del sensor DHT22. Fuente: (El Autor).

e.1.6 ANEMÓMETRO Y VELETA

El anemómetro y la veleta forman un dispositivo analógico pasivo, que nos sirve para medir la velocidad y la dirección del viento, estos dos dispositivos pueden estar juntos como también separados pero la función seguirá siendo la misma. En la figura 20 podemos ver que el anemómetro está compuesto de tres copas, con sus respectivos ejes colocados a presión en un tubo central que está en posición vertical, estas se moverán de acuerdo a la velocidad con que llegue el viento a las mismas y la veleta girará de acuerdo en la dirección que el viento llegue hacia ella, de esta manera podremos saber cuál será la velocidad y la dirección a la que se está moviendo una masa de aire.

Estos dos sensores responden a un breve pulso de excitación en un determinado movimiento. Los pulsos de velocidad son creados por un interruptor magnético sellado. “De hecho, todo el conjunto es una unidad sellada y no se pretende que se le de mantenimiento, más allá de una realineación de las piezas externas” (3).



Figura 20: Anemómetro y veleta. Fuente: (El Autor).

La conexión de estos dispositivos tienen en común su alimentación de voltaje donde los cables amarillo y rojo son Vcc y GND respectivamente, para la velocidad de viento el sensor reed switch envía sus datos al Arduino mediante el cable negro, mientras que los datos del sensor para determinar la dirección del viento se lo hace por el cable verde. En la figura 21 se presenta lo descrito.

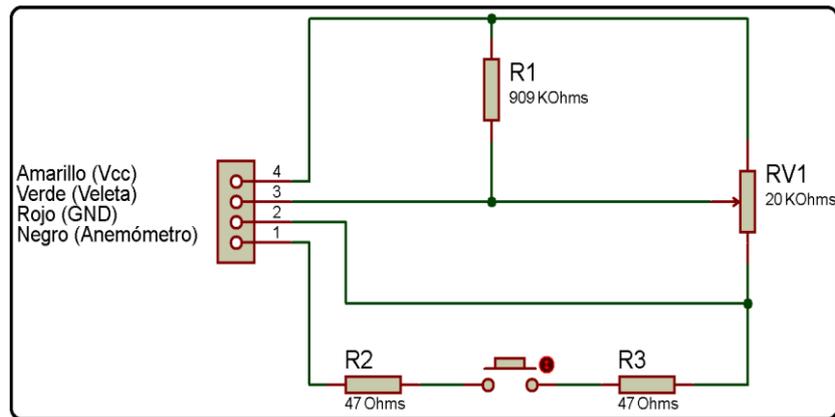


Figura 21: Esquema del Anemómetro y la Veleta. Fuente: (El Autor).

La figura muestra un interruptor magnético que se encuentra entre las dos resistencias de 47Ω , cumpliendo la función de protección al reed switch en caso de producirse un error al conectar.

Tabla de características:

Tabla 5: Características técnicas del anemómetro y la veleta. Fuente: (El Autor).

Descripción	Característica
Voltaje de operación	5V
Temperatura de operación	-40°C a +65°C
Precisión Anemómetro	± 2 mph (3 km/h, 1 m/s) o $\pm 5\%$
Rango de operación anemómetro	1 a 200 mph, 0.5 a 89 m/s, 1 a 322 km/h
Periodo de muestra de la velocidad del viento.	2.25 segundos
Precisión Veleta	$\pm 3^\circ$
Resolución de la veleta	16 puntos de 22.5° (360°)
Periodo de muestra de la dirección del viento.	1 segundo

Para la conexión de estos sensores se utiliza un circuito adicional compuesto por dos condensadores y una resistencia como se muestra en la figura 22, se describió anteriormente tiene un interruptor magnético que registra las variaciones que se producen de un nivel alto a un nivel bajo, la resistencia de $4.7K\Omega$ mantiene el interruptor en un estado en alto, por lo que cada vez que las copas giran y completan una vuelta el interruptor registra como un evento en el microprocesador ATmega328. El capacitor de $22nF$ o C1 realiza un filtrado de la señal para evitar falsas lecturas por rebote cuando se cierra el interruptor.

La Veleta es un potenciómetro con valor de $20K\Omega$; cuyo valor actual se lee en una entrada analógica del Arduino Uno y posteriormente se realiza una conversión ADC para su interpretación; cuyos saltos producidos a nivel digital son de 22.5° . También tenemos un capacitor de $10\mu F$ que realiza un filtrado de la señal de salida de la Veleta. Para mayor información técnica dirigirse a la sección de los anexos.

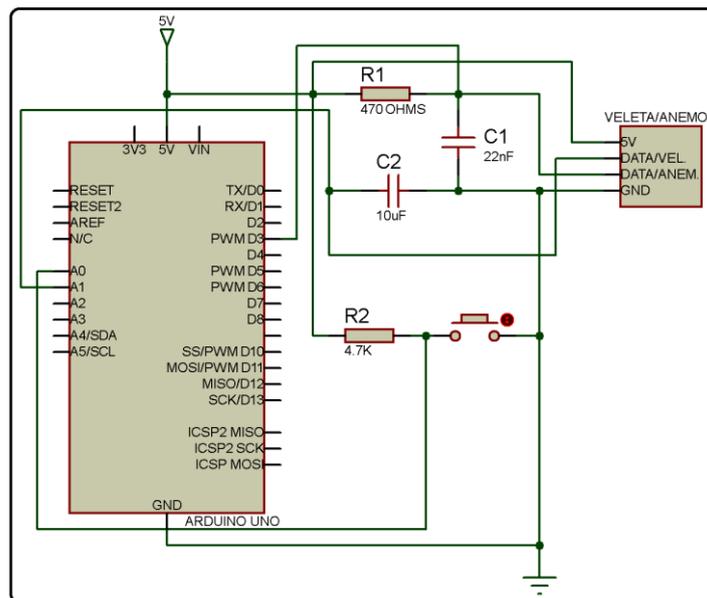


Figura 22: Esquema de conexión del anemómetro y veleta. Fuente: (El Autor).

e.1.7 PLUVIÓMETRO

El pluviómetro está compuesto por un cubo o embudo de plástico o metálico donde se reposa el agua antes de pasar por un agujero de 2 mm, con una parte afilada para cortar la lluvia en pequeñas gotas.

Existen dos tipos de pluviómetros que son:

- ✓ Con capacidad de registrar datos.
- ✓ Sin capacidad de registrar.

El primero tienen un balancín, un reed switch e imán, pero la diferencia radica en que tiene capacidad de registrar y requiere una alimentación adicional además de un microprocesador para su registro y almacenamiento de los datos en distintas unidades de tiempo, mientras que el segundo únicamente almacena el agua en un recipiente graduado. El proceso para un pluviómetro con la capacidad de registrar datos es el siguiente: el agua cae al cubo ingresa por el agujero de 0.2 cm cayendo en el balancín situado en el interior, cada vez que tenga una cantidad de agua de 0.20 mm el balancín dará un vuelco, produciendo un registro de una cantidad de agua y mientras más vuelcos se produzcan mayor será la cantidad registrada por el microcontrolador en unidades de milímetros de altura (mm).



Figura 23: *Pluviómetro Fuente: (El Autor)*

En la figura 23 se muestra el balancín y tornillos que nos sirven para calibrar la sensibilidad del pluviómetro. En la sección de anexos se puede informar más acerca de los parámetros técnicos del sensor.

Tabla de características:

Tabla 6: Características técnicas del pluviómetro. Fuente: (El Autor).

Descripción	Característica
Voltaje de alimentación	4 a 6V
Corriente máxima	1.5 mA.
Temperatura de operación	-40°C a +65°C
Rango de precipitación diaria	0.0 mm a 999.8 mm
Precisión	± 3%, conteo 0,2 mm - 50,0 mm por hora; ± 5%, conteo entre 50 mm y 100 mm por hora
Resolución	0.2 mm

En el figura 24 se presenta el esquema de conexión en donde se puede observar que tanto en la entrada de voltaje como a la salidas del pin de datos lleva resistencias únicamente con la necesidad de disminuir la corriente para evitar cualquier inconveniente con los dispositivos, ya que este sensor únicamente se toma en cuenta el cambio de estado producido por las interrupciones.

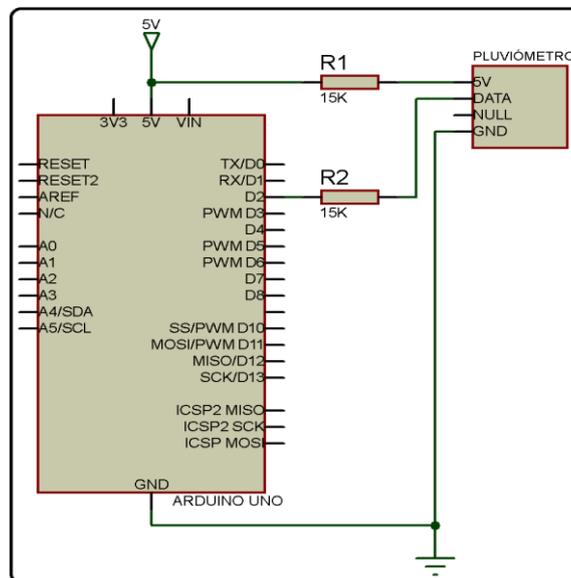


Figura 24: Esquema de conexión del pluviómetro. Fuente: (El Autor).

e.1.8 MÓDULO MICRO SD

En este proyecto se implementó un módulo SD para el almacenamiento de los datos en una memoria SD logrando de esta manera proteger los datos obtenidos por los sensores

utilizados, creando de esta manera un archivo de respaldo de dichos datos, en la figura 25 se muestra un módulo SD.

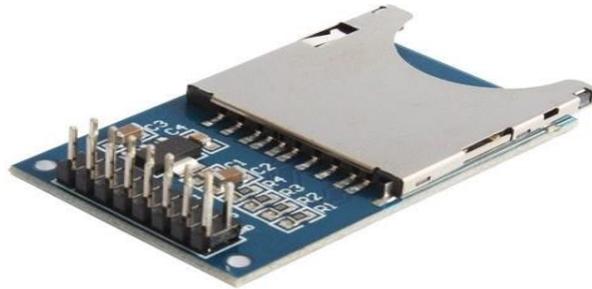


Figura 25: Módulo SD para Arduino. Fuente: (20).

Características

- ✓ Tensión de entrada 5V/3.3V.
- ✓ Corriente continua para las E/S: 20 mA.
- ✓ Pin de entrada SPI de datos: MOSI.
- ✓ Pin de salida SPI: MISO.
- ✓ Pin de activación: CS.
- ✓ Pin de clock: SCK

En la figura se muestra el esquema de conexión del módulo SD con el Arduino UNO.

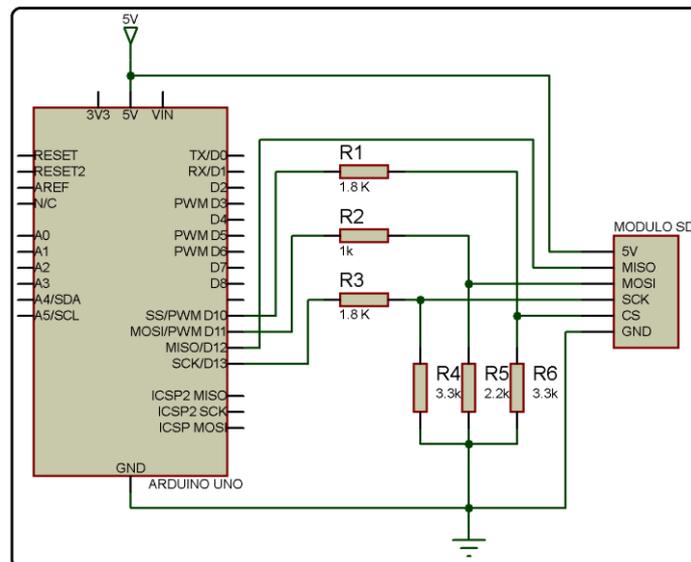


Figura 26: Esquema de conexión del Arduino con el módulo SD. Fuente:(El Autor).

e.1.8.1 MEMORIA SD

Una tarjeta SD es una memoria de almacenamiento masivo que permite registrar datos en forma segura y estable, dentro de la información que se puede almacenar tenemos videos, música, datos, etc. Estas tarjetas son empleadas en gran cantidad de dispositivos como cámaras, teléfonos móviles, video consolas, para el almacenamiento de sus respectivos datos. Sus siglas significan “Secure digital” o en español seguridad digital, este estándar fue introducido 1999 como una mejora evolutiva por sobre las tarjetas MultiMediaCard (MMC).

Secure Digital (SD) es un formato de tarjetas de memoria, en la actualidad se los utiliza en muchos dispositivos portátiles que cuenten con la ranura para su acople, como son: teléfonos, cámaras digitales, computadoras, tablet, entre otros. En la actualidad existen gran cantidad de memorias microSD de diferentes capacidades como: 1GB, 2GB, 4GB ,8GB. En el presente proyecto se utilizó una de 2GB para el almacenamiento de los datos climatológicos.

e.1.9 RELOJ EN TIEMPO REAL (RTC)

El circuito integrado DS1307 se lo utilizó por sus favorables características que se describirán a continuación:

- ✓ Bajo consumo de energía (500 nA).
- ✓ Provee información de segundos, minutos, horas, día, mes y año, la fecha final del mes se ajusta automáticamente incluyendo los meses menores de 31 días.
- ✓ Tiene formato de 24 horas o de 12 horas con indicadores AM/PM.
- ✓ Contiene 56 bytes de RAM (RAM no volátil) para almacenamiento de datos.
- ✓ Rango de temperatura industrial operacinal: -40°C a +85°C.
- ✓ Interfaz serie I2C.
- ✓ Detector automático de fallo de energía y circuito de conmutación.

En la figura 27 se muestra el CI DS1307, que se utiliza para el temporizador de Real Time Clock en el proyecto.

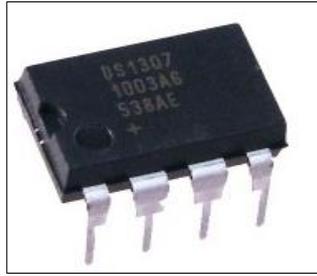


Figura 27: Circuito integrado DS1307. Fuente: (El Autor).

Al DS1307 se lo puede describir como un dispositivo de 8 pines con sus respectivas funciones, el pin 7 funciona como una salida de colector abierto, que oscila a 1Hz. Los pines 1 y 2 son utilizados para la conexión de un cristal de cuarzo estándar de 32.768 kHz. Los pines 5 y 6 que son: SDA un pin de entrada/salida para la interfaz serie de 2 hilos y SCL es el pin utilizado para sincronizar el movimiento de los datos en la interfaz serie. En la figura 28 se muestra la numeración de los pines con su respectiva función, para mayor información acerca del RTC ver en la sección de anexos.

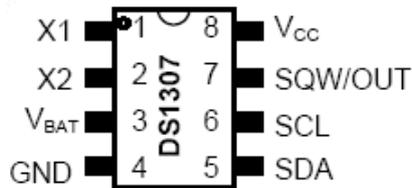


Figura 28: Descripción de los pines del DS1307. Fuente: (El Autor).

En la figura 29 se muestra el esquema de conexión de Circuito Integrado DS1307 con sus respectivos pines de salida hacia el microcontrolador Arduino328 que corresponden a los pines A4 y A5.

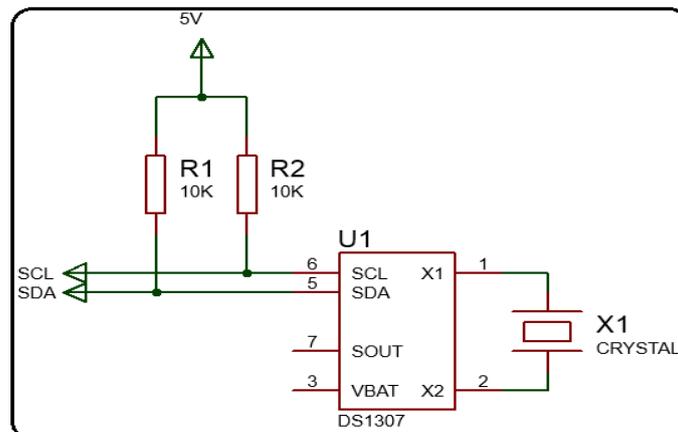


Figura 29: Esquema del RTC. Fuente: (El Autor).

e.2 MÉTODOS

e.2.1 DISEÑO DEL SOFTWARE

La estación meteorológica consta de varios sensores, que miden el estado de las variables atmosféricas, cada uno de estos tiene su propio algoritmo que cumplirá una función específica, para el cumplimiento de esta función con la mejor precisión posible se debe realizar la calibración a los sensores.

En la figura 30 se muestra la estructura principal del software implementado, la que está compuesta por bloques que especifican la cantidad de sensores a utilizarse, para la calibración de los sensores se encuentra conformado por un Arduino UNO el cual a través de instrucciones también permitirá el almacenamiento de los datos (Datalogger), este también hará la conexión a un servidor web mediante un módulo SIM900 y posterior presentación en una interfaz gráfica, donde mostrará los datos en tiempo real de las variables climáticas.

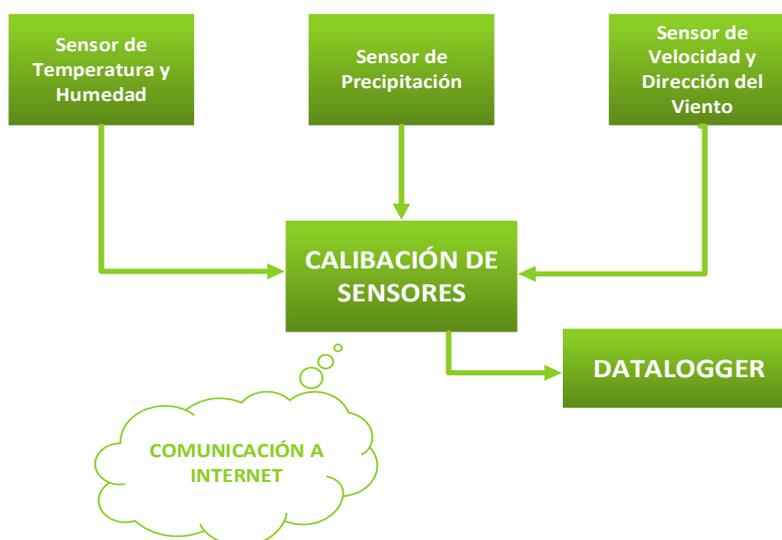


Figura 30: Estructura del software implementado. Fuente: (El Autor).

e.2.1.1 FUNCIONAMIENTO DE LOS SENSORES

En este apartado se describirá acerca de los sensores implementados y la forma en que se obtuvo los datos. En el diagrama de flujo del algoritmo implantado, primeramente se hace la asignación de los pines, luego la definición de las variables a utilizar, y la inicialización de los módulos y sensores utilizados.

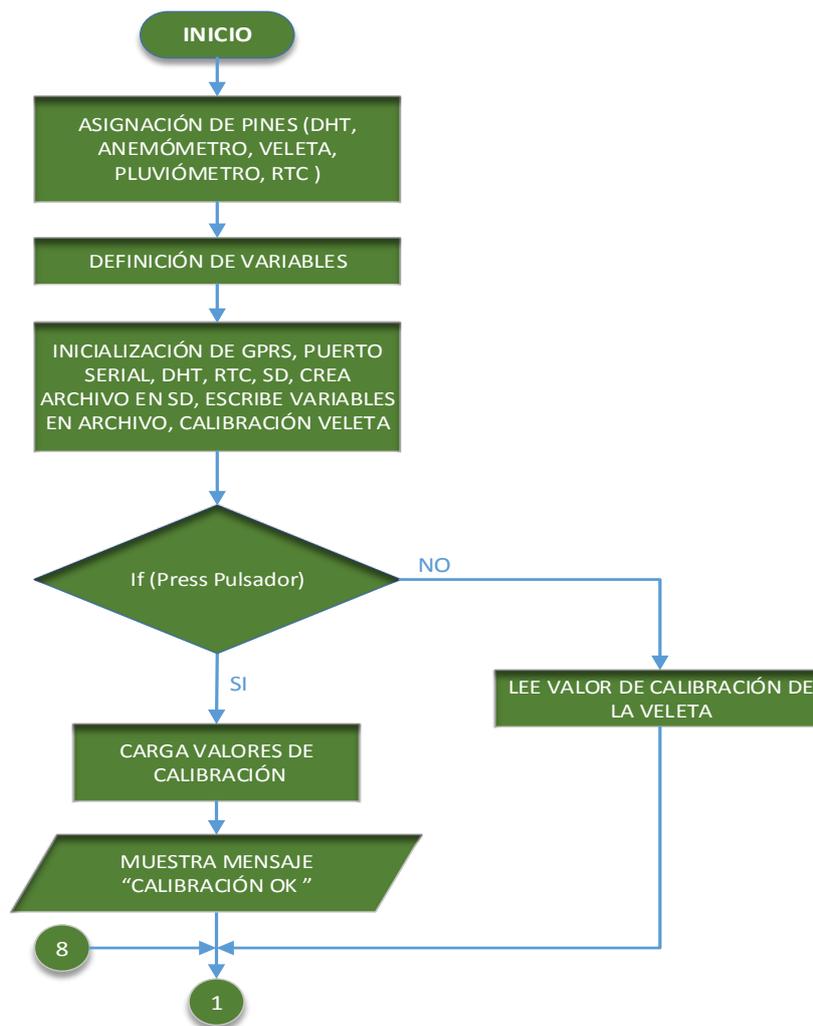


Figura 31: Diagrama de flujo del inicio del algoritmo implementado para el prototipo de estación meteorológica. Fuente: (El Autor).

e.2.1.1.1 CALIBRACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD

El sensor DHT22 es un dispositivo para medir la temperatura tal como se lo describe en la sección e.1.5, en la actualidad es muy utilizado para el desarrollo de prototipos de estaciones meteorológicas por sus características de buena calidad y una respuesta rápida en las mediciones.

Este sensor utiliza una librería que posee todos los datos de calibración realizadas en laboratorio que sirven para la medición de la temperatura y la humedad del ambiente, por lo que tiene un aceptable grado de exactitud en sus medidas; la comunicación se la realiza por un hilo o también conocido como protocolo 1-wire lo que facilita la utilización del sensor. La comunicación de este sensor consiste en enviar una señal de Start al Arduino,

como se muestra en la figura 32 se establece un nivel bajo durante 18ms y un nivel en alto de 20-40us, a continuación se configura el pin como entrada y el sensor responde con un ciclo completo pasando de un nivel bajo a un nivel alto, cada uno con una duración de 80us; el sensor envía 5 bytes de información de manera continua, además se considera el primer bit de llegada como el más significativo.

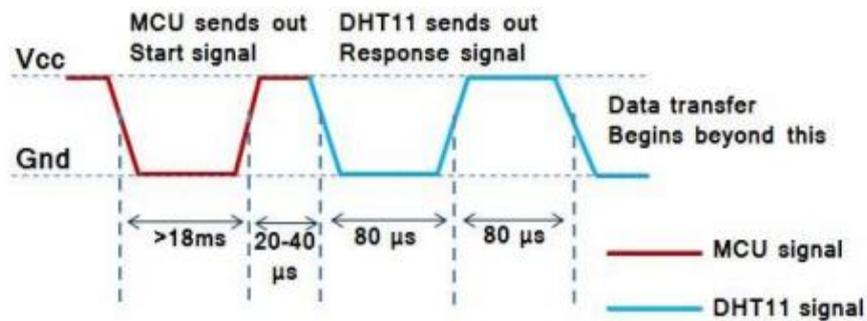


Figura 32: Señal de Start. Fuente: (21)

Los 5 bytes recibidos serán los siguientes:

- ✓ Byte1: parte entera de humedad relativa.
- ✓ Byte2: parte decimal de humedad relativa.
- ✓ Byte3: parte entera de temperatura.
- ✓ Byte4: parte decimal de temperatura.
- ✓ Byte5: check-sum.

De donde obtenemos la ecuación:

$$DATA = 16 \text{ BITS } HR + 16 \text{ BITS } TEMP. + 8 \text{ BITS } CHECK - SUM \quad Ecu \ 1$$

Donde:

$$HR = \frac{\text{valor de bits en decimal}}{10} \quad Ecu. \ 2$$

$$TEMP. = \frac{\text{valor de bits en decimal}}{10} \quad Ecu. \ 3$$

El check-sum se utiliza para confirmar que la información recibida es correcta, y se calcula sumando los 4 bytes anteriores y quedándonos sólo con los 8 bits menos significativos del resultado (21). En la figura 33 se presenta el diagrama de flujo del sensor implementado para este prototipo.

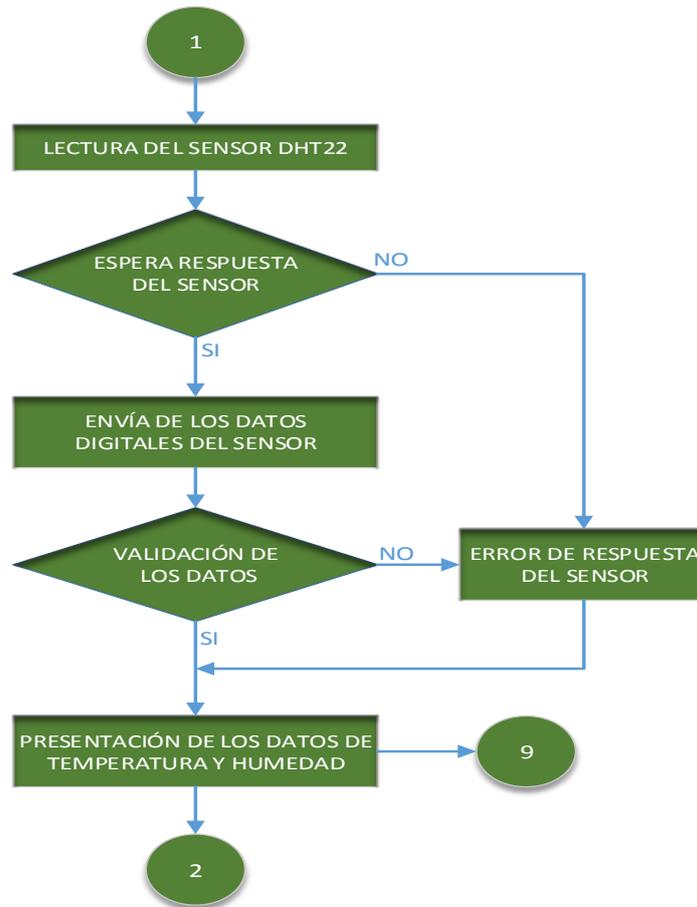


Figura 33: Diagrama de flujo de la temperatura y humedad. Fuente: (El Autor).

e.2.1.1.2 CALIBRACIÓN DE LOS SENSORES DE VELOCIDAD Y DIRECCIÓN DEL VIENTO

Para la medición de los parámetros de velocidad y dirección del viento se utiliza dos sensores como son: el anemómetro y la veleta, que en algunos casos se encuentran integrados en un solo dispositivo dependiendo del fabricante, en este proyecto se utilizó uno, marca DAVIS que vienen integrados.

El anemómetro al igual que la veleta es un dispositivo pasivo como se ha explicado en la sección e.1.6, y consiste básicamente en un interruptor magnético el mismo que por defecto se encuentra en un estado digital alto y únicamente cambia de estado cuando las copas dan una vuelta completa y el interruptor se cierra.

La salida del anemómetro va conectado al pin digital 3 del Arduino UNO para el conteo de la velocidad, se lo realiza mediante una interrupción a través de la función attachInterrupt que va asociado, la función para el conteo de la velocidad (“rpm”) a la

interrupción producida en el arduino a través del pin asignado; acto seguido el anemómetro gira una vuelta completa y produce un cambio de estado de un nivel lógico alto a un nivel bajo, iniciando así el conteo de cambios de estado para luego realizar una sumatoria y la posterior presentación del dato que definen la velocidad del viento. La ecuación para determinar la velocidad del viento es la siguiente:

$$V = P \left(\frac{2.25}{T} \right) \quad \text{Ecu. 4}$$

V = Velocidad del viento.

P = Número de vueltas.

T = Periodo de muestra en segundo.

En la figura 34 se presenta el diagrama de flujo del sensor implementado para este prototipo.

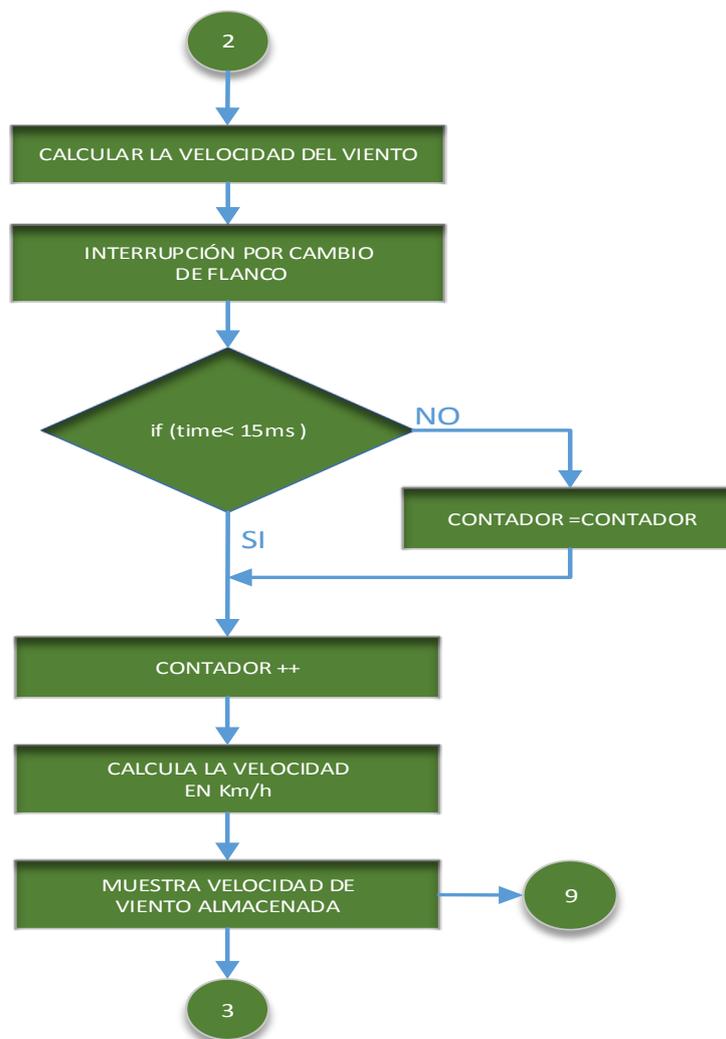


Figura 34: Diagrama de flujo de la velocidad del viento. Fuente: (El Autor).

Para la calibración del sensor de dirección del viento se lo realiza conectándolo al pin A0 donde el valor ingresado se transforma en un valor digital siendo esta una característica del Arduino como se había mencionado anteriormente, la veleta es una resistencia variable, que tiene un rango que se mapea desde el valor de 0 a 1023 (digitalmente) ya que este es un valor posible dentro del convertidor análogo digital que posee el Arduino.

Cabe mencionar que el valor de cero se toma como referencia para ubicar el Norte. Para colocar en el valor antes mencionado se hace uso de un circuito divisor de voltaje clásico como se muestra en la figura 35 cambiando de un estado alto (5V) a bajo (0V), al ubicarlo en esta posición de 0 y a partir de estos datos los demás valores se mapean y se les asigna un valor con respecto a la ubicación del desplazamiento del potenciómetro.

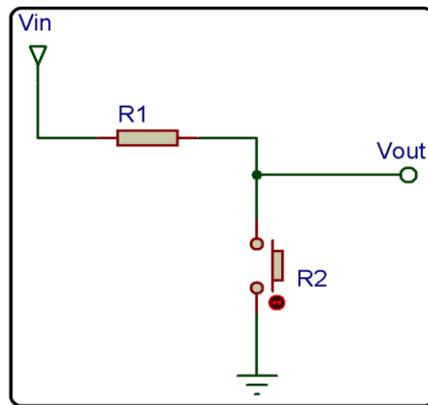


Figura 35: Circuito para calibrar la veleta. Fuente: (El Autor).

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in} \quad Ecu. 5$$

La ecuación se desarrolla en base a los valores de $V_{in} = 5V$, $R_1 = 10000\Omega$ y R_2 tendrá un valor infinito mientras el pulsador este abierto por lo que alimentará con un voltaje de 5V al pin A0 del Arduino, mientras que si presionamos el pulsador la resistencia será cero por lo que llegará al pin A0 un nivel bajo (0V), determinando la dirección norte como se explicó anteriormente. El diagrama de flujo que representa el algoritmo desarrollado se muestra a continuación:

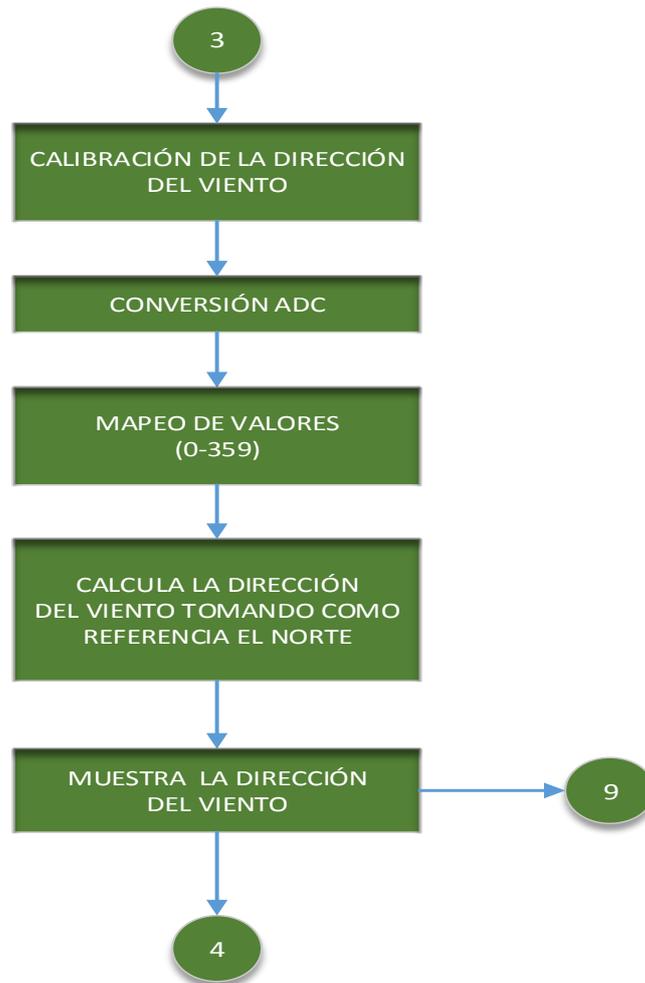


Figura 36: Diagrama de flujo de la dirección del viento. Fuente: (El Autor).

e.2.1.1.3 CALIBRACIÓN DEL SENSOR DE PRECIPITACIÓN

El pluviómetro es un sensor que está destinado a medir la cantidad de agua caída, en un proceso de lluvia, para la medición de este parámetro se utiliza un reed switch que detecta el cambio de un flanco de bajada; la salida de este sensor va conectado al pin digital 2 del Arduino UNO; es bastante parecido al anemómetro ya que utiliza una interrupción para el conteo de los cambios de estado y por ende hace uso de la función attachInterrupt, lo que hace que cada que se registre un vuelco del balancín el contador vaya incrementando su valor hasta llegar a un intervalo de tiempo de reinicio. Esta variable meteorológica se obtiene mediante la ecuación 6, donde se toma en cuenta el número de pulsos o de cambios de estado que produce el balancín multiplicado por el factor de resolución que nos ofrece la hoja de datos de dispositivo.

$$X = \text{cambio_puls} \times 0.20\text{mm} \qquad \text{Ecu. 6}$$

Donde:

X = cantidad de lluvia.

$Cambio_pluv$ = conteo de los cambios de estado.

0.20 = constante de calibración.

Para la calibración por hardware se tiene dos tornillos pequeños en la base del pluviómetro de marca DAVIS que al girarlos a una determinada dirección se hace variar el nivel de agua que recoger en cada copa del balancín (0.01 a 0.2 mm) antes de producirse un vuelco, es recomendable dejar en la posición que llega de fábrica ya que viene calibrado para la medida que muestran en su hoja de datos, por lo que cabe decir que está calibrado con una medida de 0.2 mm por copa. En la figura 37 se presenta el diagrama de flujo del sensor implementado para este prototipo.

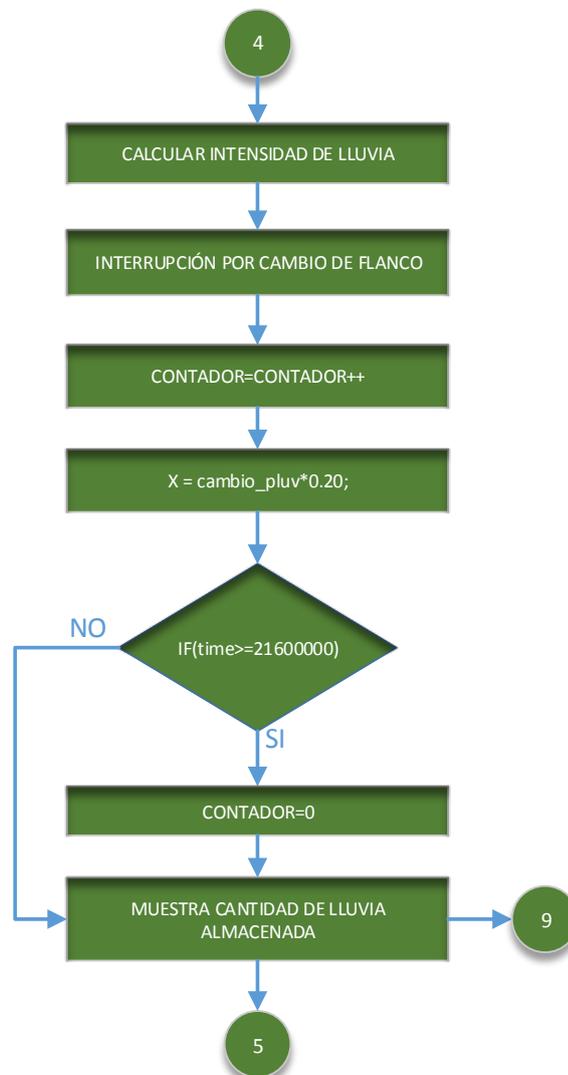


Figura 37: Diagrama de flujo de la intensidad de lluvia. Fuente: (El Autor).

e.2.1.2 DATALOGGER

Las estaciones meteorológicas están compuestas de un Datalogger que almacena los valores obtenidos por los sensores en determinados instantes de tiempo, siendo este un respaldo de los datos por si llegara a ocurrir alguna falla de cualquier índole que hiciera que este dispositivo pierda los datos almacenados en el servidor, al que estará transmitiendo o su transmisión dejare de funcionar; estos datos nos dan a conocer las variaciones que se producen en el clima, que nos informara o alertara de cualquier eventualidad.

Para el almacenamiento de los datos en una SD, es de vital importancia tener en cuenta un factor como es el tiempo en que suceden cada una de las variaciones climatológicas, para lo cual se hace necesario contar con un RTC o llamado también reloj en tiempo real, donde la principal función es mostrar la fecha y hora actual tomando como referencia la del computador. Para iniciar este proceso primeramente se debe inicializar el dispositivo, para luego proceder a crear un archivo y escribir un encabezado dentro del mismo para luego escribir las variaciones climatológicas con la fecha y hora determinada de cada medición.

Para la conectividad de estos dispositivos con la placa Arduino se posee una librería, misma que hace la comunicación entre estos dos dispositivos. En la figura 38 se presenta el diagrama de flujo para el almacenamiento de los datos incorporado en este prototipo.

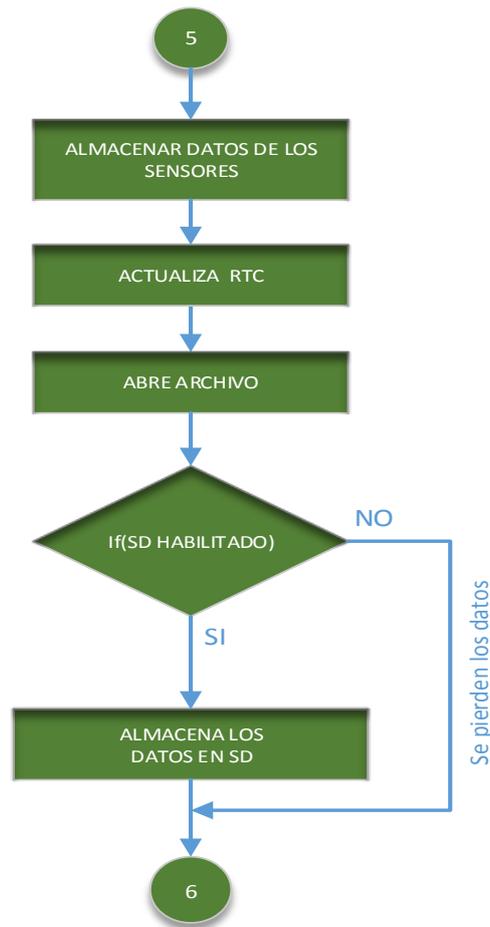


Figura 38: Diagrama de flujo para el almacenamiento de los datos generados por los sensores. Fuente: (El Autor).

El formato empleado para el almacenamiento de las variables climatológicas se dará a conocer en la siguiente tabla:

Tabla 7: Presentación de los datos de cada una de las variables almacenadas en la SD. Fuente: (El Autor).

Temperatura	Humedad	Vel_del viento	Direc_del viento	Precipitación	Hora	Fecha
24.60	53.00	2.40	112	0.00	14:25	2014/9/12
24.80	52.90	1.20	338	0.00	14:25	2014/9/12
24.80	52.10	2.40	337	0.00	14:26	2014/9/12
24.90	52.00	1.20	340	0.00	14:26	2014/9/12
25.00	51.80	2.40	342	0.00	14:27	2014/9/12
25.00	51.90	4.80	30	0.00	14:27	2014/9/12

Los datos de cada una de las variables de temperatura, humedad, velocidad y dirección del viento, precipitación, son almacenadas en intervalos de tiempo de 30 segundos tal como se presenta en la tabla garantizando de esta manera una cantidad de 120 muestras por hora las mismas servirán en caso de que se use para realizar las respectivas gráficas, tomando cualquier programa que crea conveniente para hacerlo.

e.2.2 COMUNICACIÓN A INTERNET

La estación meteorológica establece la comunicación a internet a través de un módulo SIM900 GSM/GPRS para enviar los datos de las variables climáticas que se tomaron en consideración, para esta comunicación se estableció la arquitectura cliente/servidor, entre el Arduino Uno y el servidor “emoncms.org”.

Para tener acceso a este servidor se debe crear una cuenta, para esto cabe recalcar que existen dos tipos de cuentas con costo y la otra gratuita, para este proyecto se utilizó la cuenta gratuita, para tener acceso a esta plataforma y poder realizar las configuraciones necesarias para la presentación de los datos de cada una de las variables consideradas.

El Arduino UNO es la parte fundamental dentro de este proyecto ya que aquí es donde se configuran los parámetros para ingresar a la red de internet por medio del módulo GSM/GPRS y se envía los datos en una trama al servidor mediante el protocolo HTTP.

e.2.2.1 CONFIGURACIÓN DE LA RED

El desarrollo de esta configuración es muy importante ya que puede cambiar de acuerdo al módulo de transmisión que se utilice, pueden ser por configuración Ethernet, wifi, GSM/GPRS, entre otros. En este caso se utilizó un módulo SIM900 GSM/GPRS, ahora la placa Arduino será la encargada de establecer la comunicación mediante comandos AT para lo cual se debe establecer o comprobar algunos parámetros como: comprobar que el reporte de calidad de señal sea la adecuada ("AT+CSQ"), verificar si el chip está conectado a la red ("AT+CGATT=1"), configurar el módulo para aplicaciones basadas en IP con el comando "AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\" donde los números 3 y 1 ajustan los parámetros y habilitan la conexión con el portador mientras que CONTYPE nos da a conocer el tipo de conexión, configurar el nombre del punto de acceso, de acuerdo al operador que se utilice con el comando

e.2.2.2 CREACIÓN DE UNA INTERFAZ PARA EL REGISTRO DE DATOS

El uso de servidores gratuitos para el desarrollo de proyectos ha facilitado en gran parte la presentación de los mismos, cabe aclarar que al hacer uso de estos servidores debemos sujetarnos a políticas de los propietarios de dicha plataforma que ofrecen este servicio, en este proyecto se utilizó el servidor emoncms.org, siendo este una muy buena herramienta para el desarrollo de aplicaciones web, ya que permite procesamiento, manipulación y visualización de datos principalmente de energía, pero brinda la facilidad de poder transmitir datos de sensores climatológicos (temperatura, humedad, velocidad y dirección del viento, intensidad de lluvia). Este servidor está desarrollado con código abierto lo que hace que sea mucho más eficiente ya que se puede hacer algunas modificaciones para su personalización.

Es muy importante conocer algunas de las reglas, que son de mayor importancia y que son establecidos por los desarrolladores del servidor antes mencionado, dentro de estas tenemos: el tiempo que se pueden enviar datos no deben ser inferior a 5 segundos, el tiempo de actualización de los datos en el servidor debe ser mayor o igual al tiempo que se establece en la transmisión desde el Arduino, los datos publicados son propiedad de quien los crea, con la principal condición de no realizarlo con fines comerciales.

El código fuente de este servidor se encuentra en el servicio de hosting basado en la web “GitHub” donde los desarrolladores de códigos de “open source” trabajan con las versiones y contribuyen en el desarrollo de diferentes códigos. Emoncms es un Sistema de Gestión de Contenidos (CMS), por lo tanto es un sitio web con una interfaz gráfica de usuario que se compone de una serie de módulos apoyándose en una base de datos MySQL para almacenar dichos datos y las diversas opciones de la propia interfaz gráfica de usuario. La interfaz de usuario es totalmente personalizable, y adaptable (13).

Emoncms nos permite llevar el monitoreo de variables meteorológicas de una manera fácil, rápida y eficiente, como cualquier servidor tiene un menú de configuraciones donde tenemos cuatro parámetros muy importantes como son: input, Feeds, Vis y los Dashboard, los mismos que realizan una función específica. A continuación describiremos las funciones de cada uno de los parámetros.

Inputs.- Aquí es donde se almacenan las variables de los sensores utilizados que se controlaran de manera remota, esta plataforma permite realizar una calibración adicional para optimizar el procesamiento de los datos y obtener los mismos con mayor precisión posible.

La configuración de las entradas se la realiza mediante tres procesos fundamentales, donde se asigna un nombre al o los procesos que se esté llevando a cabo, también se le da un intervalo para el registro de los datos, aquí es importante colocar un valor mayor o igual al intervalo dado en el proceso de transmisión de los datos, y por último se realiza la asignación de un proceso como realizar una suma, resta o también colocar el valor máximo o mínimo de los datos que se transmiten, y que como administrador de la cuenta se puede configurar. En la figura 40 se pueden observar los parámetros que se deben configurar.



Node:	name	Description	Process list
1	3		
1	2		
1	1		

Figura 40: Entorno de las entradas de emoncms.org. Fuente: (22).

- ✓ **Node.-** Hace referencia al dispositivo que envía los datos al servidor (pueden contener distintas variables).
- ✓ **Name.-** El sistema no sabe qué tipo de datos está recibiendo entonces se debe proporcionar un nombre que se le asigna a la entrada: por ejemplo, temperatura, humedad, velocidad del viento, dirección del viento intensidad de lluvia o precipitación, entre otras.
- ✓ **Description.-** se realiza una descripción a las variables que se encuentra monitoreando.
- ✓ **Process list.-** Muestra el tipo de procesamiento que se le haya asignado a cada variable de entrada.

Feeds.- Es donde se gestiona el almacenamiento de los datos de entrada, aquí se puede ver la actualización de cada variable de acuerdo al tiempo configurado para su actualización en el servidor, dentro de este campo tenemos ID que corresponde al identificador de un determinado proceso, el mismo que más adelante será utilizado para crear la interfaz gráfica de la variable sensada y se ocupa de pasarlos a los siguientes parámetros. Siempre dentro de este parámetro, se puede visualizar la actividad de envío de datos, tiene algunos campos, el más importante de estos es el ID que corresponde al identificador del proceso, existe otro campo que es “Public” este se configura dependiendo si se desea que los datos sean públicos para que puedan ser vistos por cualquier personas, o simplemente privados que limitarán el acceso a una cuenta segura. La figura 41 podemos observar en detalle las opciones que presta el servidor dentro del parámetro Feed.

Id	Name	Tag	Datatype	Public	Updated	Value			
155	power	house	REALTIME	🔒	5s ago	100.0	/	🗑️	⚙️
156	kwhd	house	DAILY	🔒	5s ago	0.00	/	🗑️	⚙️
157	histogram	house	HISTOGRAM	🔒	5s ago	100.0	/	🗑️	⚙️

Figura 41: Entorno del campo “Feed” de emoncms. Fuente: (22).

Vis.- Este parámetro es donde se realiza las visualizaciones de cada una de las variables transmitidas, donde se configura de manera gráfica la selección de la visualización, una vez que se visualiza la gráfica deseada, también presenta un código html de la misma, del que podemos hacer uso para colocarla en una página web con o sin dominio y poder presentarla en tiempo real. El proceso completo para la visualización de una gráfica se describe a continuación:

- ✓ **Select visualisation.-** En la pestaña que se encuentra a la izquierda de la ventana escogemos el parámetro en el cual se está transmitiendo las variables, en este caso es en tiempo real.
- ✓ **Set options.-** En la pestaña que se encuentra frente a FeddID, escogemos la variable a la que queremos visualizar.

- ✓ **View.-** Al dar click en el icono view nos muestra la gráfica de la variable que llamamos o que deseamos monitorear.
- ✓ **Full screen.-** Nos permite mostrar la gráfica en pantalla completa.

En la parte inferior nos muestra un código html que se describió anteriormente del mismo que se puede hacer uso para publicarla en una página web. En la figura 42 se aprecia lo descrito acerca de cada uno de los parámetros.

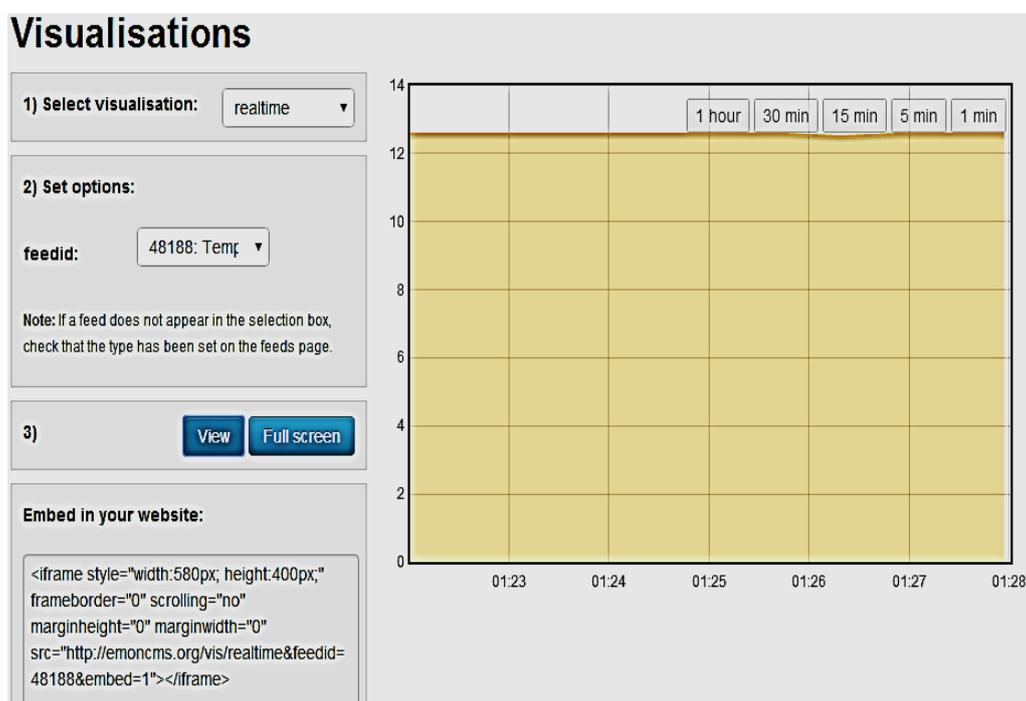


Figura 42: Visualización de variables en emoncms.org. Fuente: (El Autor).

Dashboard.- El Dashboard es la ventana en donde se diseñará la interfaz gráfica, la misma que se presenta para la visualización de las variables, existen tres tipos de configuraciones para la presentación de esta interfaz, cada uno de estos con características distintas, las mismas que describiremos a continuación:

- ✓ **Main page.-** Esta configuración de página nos permite mostrar la aplicación diseñada, únicamente con escribir siguiendo el formato de la siguiente dirección electrónica “emoncms.org/Usuario”.
- ✓ **Page Published.-** Estas páginas son de libre acceso para que sea visitada por cualquier persona, para observar los datos presentados sin opción a editar los mismos.

- ✓ **Page Private.-** Para acceder a estas páginas debe autenticar su cuenta en el servidor emoncms.org.

Para el desarrollo de la interfaz hacemos uso de la barra de herramientas que nos presta en esta ventana dentro del cual tenemos los siguientes:

- ✓ **Text.-** Esto nos permite escribir un texto en el editor de dibujo, existen varios tipos: título centrado, subtítulo, párrafo, con distintos tamaños de letra.
- ✓ **Widgets.-** Permite realizar una animación en tiempo real de los cambios de las entradas, dentro de los cuales tenemos los dial, jgauge, feed value entre otros.
- ✓ **Visualisations.-** Las visualizaciones se pueden configurar de varias maneras, dependiendo de los requerimientos gráficos se los puede presentar en tiempo real, gráficos del valor de los datos, en gráficos de estadísticas diarias, mensuales y anuales, histogramas, multigráficas, entre otros.
- ✓ **Configure.-** Este campo nos permite configurar nuestros widgets, seleccionando el widget que se requiera configurar y luego se establecen algunos parámetros, como unidades de acuerdo a las variables que se estén midiendo.

El desarrollo del diseño para el prototipo de la estación meteorológica se lo realizó utilizando text, widgets (feedvalue, jgauge), visualization (realtime, simplezoom). Para utilizar el text tenemos tres variantes que son texto para título, subtítulo centrado, para texto de párrafo en este caso se hizo uso de los tres tipos de texto, seguidamente tenemos los widgets en donde se muestran las gráficas o valores que son prediseñados como el feedvalue nos permite adquirir y visualizar el valor numérico de un dato de entrada, el jgauge, permite tener una animación en tiempo real de los cambios que se generan en la entrada del dato que se desee leer, luego tenemos la visualization en donde nos mostrará la gráfica respectiva de la variable que vayamos a medir en función del tiempo, en la mayoría de los casos se usa en tiempo real, o también promedios diarios, semanales, mensuales, anuales. Todas las variables medidas en este proyecto se presentarán en gráficas de tiempo real con actualizaciones cada 30 segundos, las mismas que son: temperatura, humedad relativa al aire, la velocidad y dirección del viento, la precipitación o intensidad de agua caída. Una característica importante de la gráfica es que podemos ver los valores en intervalos de una hora, 30 minutos, 15 minutos, 5 minutos o en 1

minuto. Para verificar lo descrito se muestra en la figura 43 las variables de temperatura y humedad del ambiente.

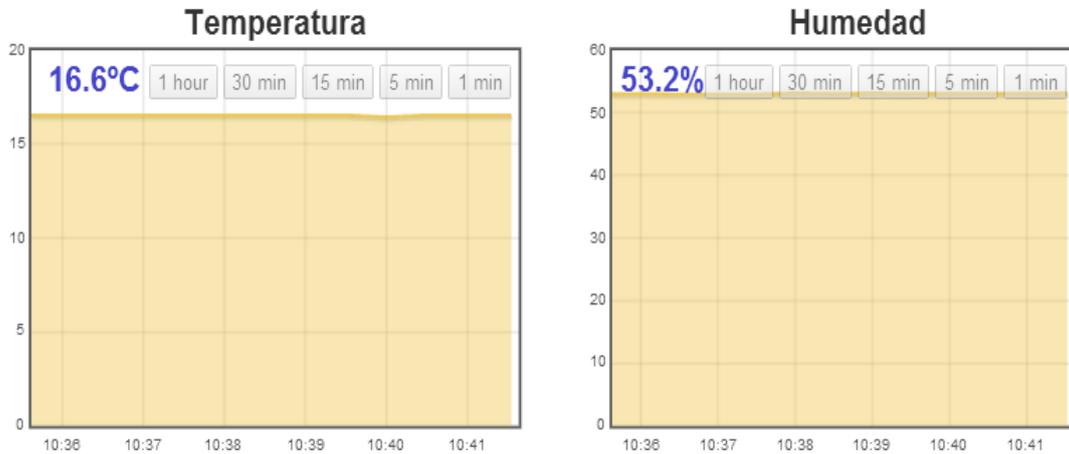


Figura 43: Interfaz gráfica para monitorear temperatura, humedad, en tiempo real.

Fuente: (El Autor).

También se puede graficar las variables de temperatura y la humedad en el lapso de días, semanas, meses o años, donde se visualizan las variaciones producidas en estos determinados intervalos de tiempo. La interfaz con estas dos variables se muestra en la siguiente figura.

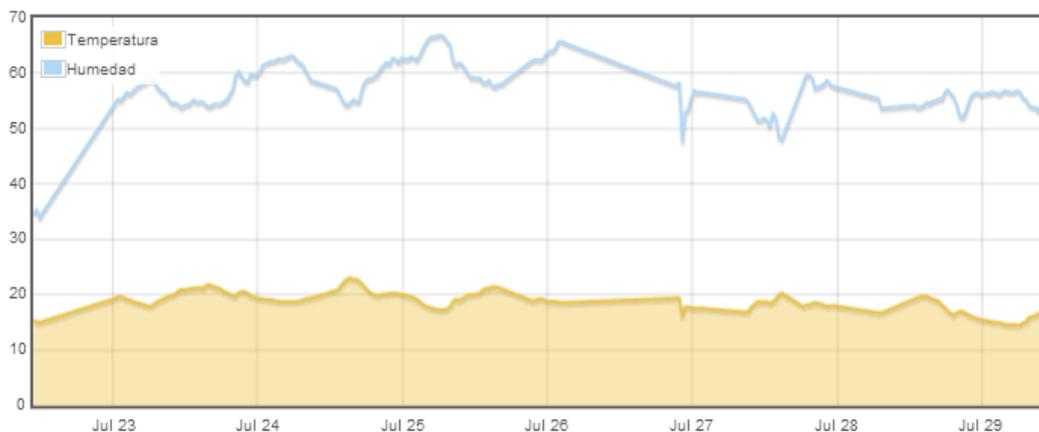


Figura 44: Interfaz gráfica para el monitoreo de la temperatura y la humedad. Fuente:

(El Autor).

La velocidad y la dirección del viento igualmente se visualizan en intervalos de 60, 30, 15, 10, 5, 1 minuto, en el cual se puede observar las variaciones producidas por los sensores. La interfaz con estos dos parámetros se muestra en la figura 45 con actualizaciones cada 30 segundos.

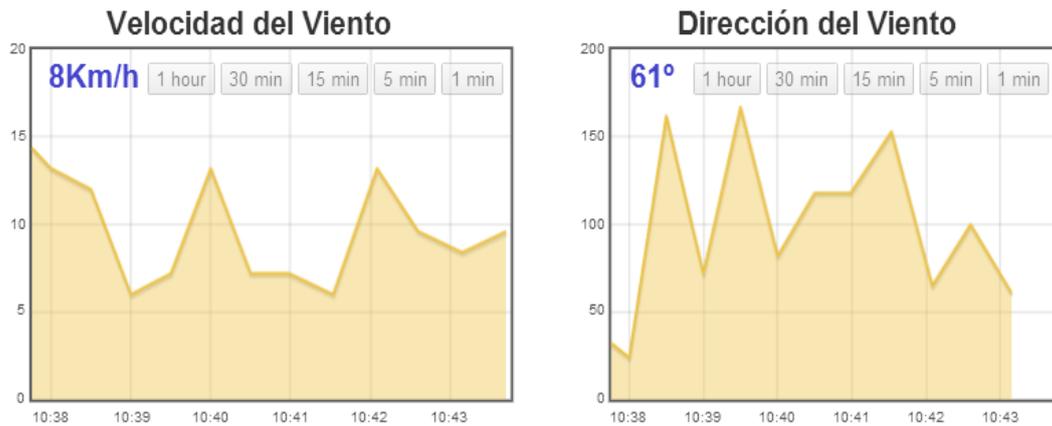


Figura 45: Interfaz gráfica para el monitoreo de la Velocidad y la dirección del viento en tiempo real. Fuente: (El Autor).

Por último se mostrará la precipitación en tiempo real y también en lapsos de días, semanas, meses y años, para poder visualizar las variaciones producidas en cualquiera de las opciones que se nombraron anteriormente. La interfaz con estos parámetros se muestra en la figura 46.



Figura 46: Interfaz gráfica para el monitoreo de la precipitación. Fuente: (El Autor).

e.2.3 SISTEMA DE CONTROL DE VERSIONES GITHUB PARA PRESENTACIÓN DE PÁGINA WEB.

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. Como se encuentra fundamentado en la sección d.3.2 Linus Torvalds diseño GitHub, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Entre las características más relevantes se encuentran:

- ✓ Git mantiene una enorme cantidad de código distribuido y gestionado por mucha gente, que incide en numerosos detalles de rendimiento.
- ✓ Ofrecen fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclas de diferentes versiones.
- ✓ Los almacenes de información pueden publicarse por HTTP, mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH.
- ✓ Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- ✓ Realmacenamiento periódico en paquetes (ficheros).
- ✓ Adicional a esto brinda la posibilidad de importar todas las gráficas de transmisión en tiempo real a un sitio web personalizado, en el presente proyecto se creó un diseño web en base a una plantilla prediseñada y mediante la herramienta web “GitHub” se subió a la web de manera gratuita y se importaron algunas gráficas de las que se crearon en el servidor prediseñado emoncms.org.

e.2.3.1 CONFIGURACIÓN EN GITHUB

Para hacer uso del control de versiones primeramente debemos crear una cuenta como se muestra en la figura 47 la creación de esta cuenta consta de dos pasos la primera es sobre los datos como usuario, mail y contraseña en la figura 48 escogemos la cuenta que deseamos, para el proyecto escogemos la cuenta gratuita y colocamos finalizar, con estos pasos tenemos ya creada la cuenta.

Step 1:
Set up a personal account

Step 2:
Choose your plan

Create your personal account

Username
unlemeteorologica ✓

This will be your username — you can enter your organization's username next.

Email Address
@gmail.com ✓

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password
•••••••• ✓

Use at least one lowercase letter, one numeral, and seven characters.

Confirm your password
••••••••

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

Figura 47: Interfaz gráfica para la configuración de la cuenta en GitHub. Fuente: (El Autor).

Completed
Set up a personal account

Step 2:
Choose your plan

Choose your personal plan

Plan	Cost	Private repos	
Large	\$50/month	50	Choose
Medium	\$22/month	20	Choose
Small	\$12/month	10	Choose
Micro	\$7/month	5	Choose
Free	\$0/month	0	Chosen

Don't worry, you can cancel or upgrade at any time.

Help me set up an organization next
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations.](#)

Finish sign up

Figura 48: Interfaz gráfica para la configuración de la cuenta en GitHub. Fuente: (El Autor).

Se debe crear un repositorio para el alojamiento de los datos Html, con el nombre del usuario añadiendo “.github.io” para luego poder hacer públicos estos datos o poder presentar como páginas web, en la figura 49 se muestra como se crea un repositorio.

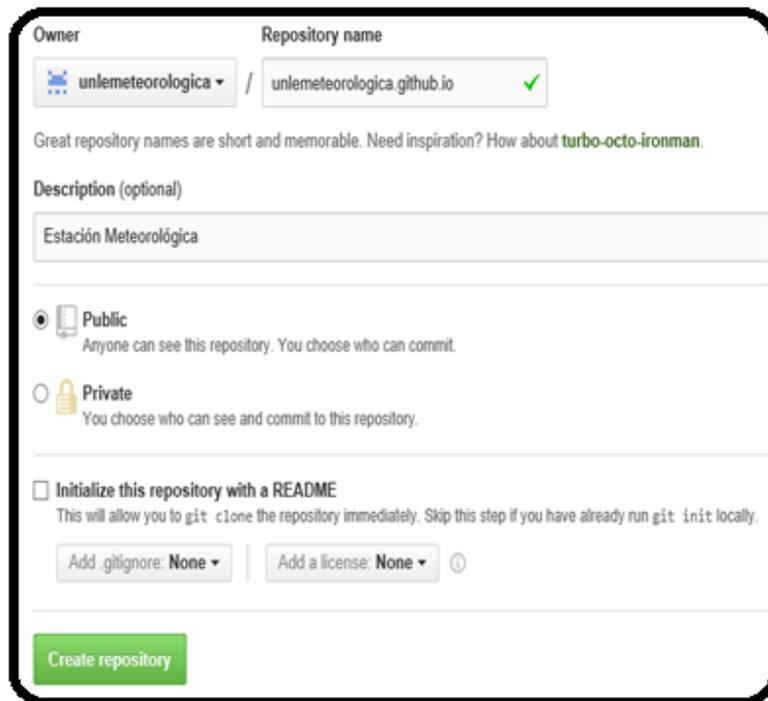


Figura 49: Interfaz gráfica para la creación de un repositorio en GitHub. Fuente: (El Autor).

Una vez creado el repositorio debemos crear una carpeta con el mismo nombre del repositorio en la cuenta del usuario master del ordenador, para ahí colocar los archivos que se vaya a subir a la página de GitHub, para esto se debe sincronizar la consola Git bash y la página de GitHub.

Git bash es una consola donde se trabajará bajo comandos, para cargar los documentos al repositorio de GitHub se debe primeramente configurar el usuario, email y luego iniciar la carga de los archivos a continuación se da a conocer todos los comandos que se utilizan para realizar este proceso.

Paso 1: Configurar cuenta

```
git config --global user.name [nombre de la cuenta ]
```

```
git config --global user.email [@.example.com]
```

Paso 2: Subir los archivos

Desde la consola bash ingresamos a la carpeta que están los archivos a cargar una vez dentro de esto colocamos los comandos siguientes para cargar los archivos cabe mencionar que se debe tener conocimientos básicos de uso de consola y de Linux.

Cd.- Sirve para entrar en el directorio o la carpeta donde tenemos nuestros archivos ya dentro del directorio deseado realizamos el siguiente proceso con ayuda de los comandos necesarios.

- ✓ `git init` // Creamos un nuevo repositorio en git
- ✓ `git add.` // Con esto se suben todos los archivos que deseamos o únicamente ponemos el nombre del archivo que deseamos cargar en vez del punto.
- ✓ `git commit -m "comentario"` // Agregamos un comentario al bloque de archivos que se carga al control de versiones.
- ✓ `git remote add origin` // Aquí se incluye una dirección https que nos da luego de crear el repositorio en GitHub.
- ✓ `git push -u origin master` // Con este comando enviamos todos los documentos seleccionados a GitHub.

Una vez realizado todo este proceso, ya tenemos cargados los archivos o documentos en GitHub. En la figura 50 se muestra la consola git bash.

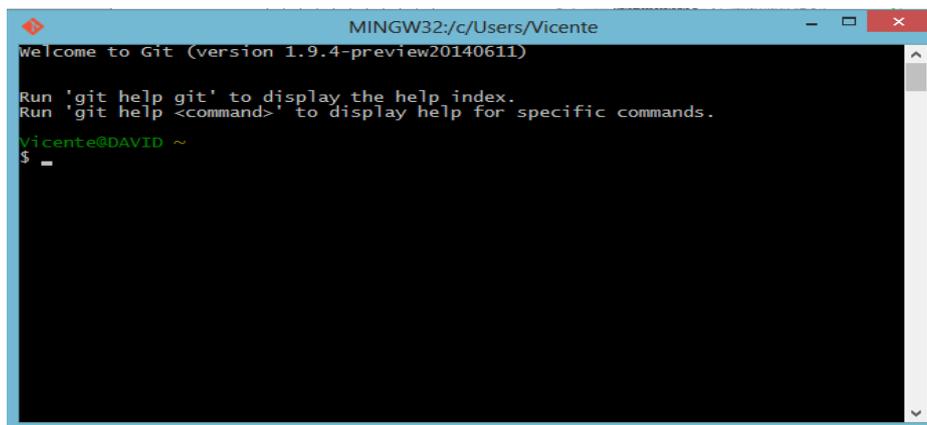


Figura 50: Interfaz de la consola git bash para la configuración de cuenta en GitHub.

Fuente: (El Autor).

Una vez cargados los archivos en GitHub a través de git bash, podemos observar en la cuenta del sistema de control de versiones, todos los archivos alojados, entre los de mayor utilidad como: las imágenes, archivos de javascripts, los documentos .html, tal como se muestra en la figura 51.

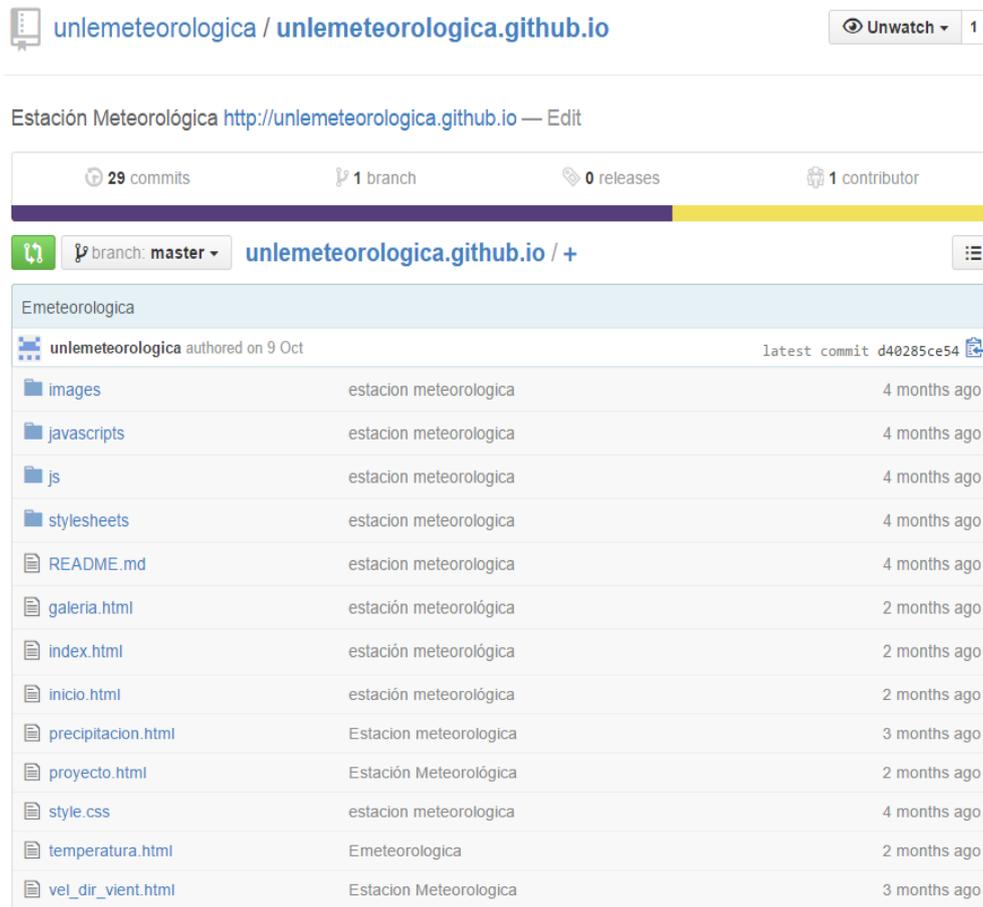


Figura 51: Presentación de archivos subidos en GitHub a través de la consola git bash.

Fuente: (El Autor).

e.2.3.2 MODELO DE PRESENTACIÓN DE LOS DATOS METEOROLÓGICOS

El modelo para la presentación se lo desarrolló en base a una plantilla predefinida a la que se le realizó un rediseño adecuándole a las necesidades deseadas, la misma que consta de 6 items que son: inicio, proyecto, temperatura y humedad, velocidad y dirección del viento, precipitación y galería.

En **inicio** hace una introducción sobre los sistemas meteorológicos, utilidad y servicios, transmisión los datos; en **proyecto** se hace una breve explicación del diseño

implementado, fases que se desarrollaron y descripción de cada una de ellas; **temperatura y humedad** primeramente describimos el sensor utilizado, sus unidades de medida correspondientes, mostramos el sensor DHT22 y las respectivas gráficas de cada una de estas variables y una comparativa entre estas; **velocidad y dirección del viento** igualmente que la anterior se describe el sensor, las unidades de medida, presentación del anemómetro y veleta, las gráficas correspondientes para saber la velocidad a la que se está moviendo, la dirección de donde proviene el viento y por último la gráfica comparativa de las dos variables; **precipitación** igualmente se da a conocer la unidad de medida, descripción del dispositivo, muestra las gráficas; por último tenemos la **galería** donde se presentan los dispositivos empleados y los videos que se realizó acerca del prototipo de estación meteorológica.

Es importante tener en cuenta que las gráficas podrán ser observadas por diferentes fracciones de tiempo, así como los historiales que se puede observar durante días, semanas, meses y años, en la figura 52 se muestra la parte principal de la página subida a través de GitHub, la misma que está disponible para que puedan observar las variables climáticas sensadas por el prototipo de estación meteorológica que está en pleno funcionamiento en el sector de la Argelia de la ciudad de Loja.



Figura 52: Imagen de la página principal del prototipo de estación meteorológica.
Fuente: (El Autor).

f RESULTADOS

Una vez realizada la construcción del hardware e implementación del software, se pone en funcionamiento la estación meteorológica para el desarrollo de las respectivas pruebas que acrediten que es o no de utilidad el diseño realizado; los datos serán transmitidos a **emoncms.org** el servidor gratuito donde se alojarán y se presentará en forma gráfica con los respectivos valores de cada variable sensada, además se podrá observar los datos en la página web con dirección electrónica <http://unlemeteorologica.github.io> o también se podrá observar en la página del servidor escribiendo la dirección <http://www.emoncms.org/VicentePoma> o ingresando a la cuenta ; pudiendo hacer uso de esta información en cualquier área que se la requiera.

Los datos de temperatura tomados por prototipo serán comparados con los de la estación meteorológica del INAMHI y los datos de velocidad y dirección del viento, precipitación serán comparadas con la estación meteorológica comercial de marca VAISALA QML210 existente en la Universidad Nacional de Loja en el Área de la Energía las Industrias y los Recursos Naturales No Renovables, dando mayor certificación del prototipo implementado; también se conocerá el margen de error que se produce entre las estaciones y así conocer las posibles variaciones que pueden existir. Además se incluye una SD para el almacenamiento y protección de los datos, por posibles inconvenientes.

f.1 SENSOR DE TEMPERATURA

En la siguiente tabla se muestran los valores adquiridos en intervalos de tiempos de tres horas iniciando a las 07H00 con la primera medida y terminando a las 19H00.

Tabla 8: Comparación de las medidas de temperatura. Fuente: (El Autor).

PROTOTIPO IMPLEMENTADO	ESTACIÓN METEOROLÓGICA "INAMHI"	DESVIACIÓN	ERROR (%)
14,00	13,00	1,00	0,08
22,00	20,00	2,00	0,10
24,00	22,00	2,00	0,09
22,00	22,00	0,00	0,00
16,50	17,00	-0,50	-0,03
13,50	13,00	0,50	0,04
25,40	20,00	5,40	0,27
25,00	24,00	1,00	0,04
15,40	16,00	-0,60	-0,04
15,30	17,00	-1,70	-0,10
		ERROR	0,05

f.2 SENSOR DE HUMEDAD

En la siguiente tabla se muestran los valores adquiridos en intervalos de tiempos de tres horas iniciando a las 07H00 con la primera medida y terminando a las 19H00. Estos datos no fueron comparados porque no existe en el sector estación que mida este parámetro.

Tabla 9: Medidas de humedad relativa. Fuente: (El Autor).

PROTOTIPO IMPLEMENTADO
98,70
64,60
50,70
64,60
96,8
99,90
61,40
61,20
98,20
98,00

f.3 SENSOR DE VELOCIDAD DEL VIENTO

En la siguiente tabla se muestran los promedios por hora de medición de la velocidad del viento; se consideraron los promedios de las siguientes horas: 07H00-08H00, 10H00-11H00, 13H00-14H00, 16H00-17H00 y 19H00-20H00.

Tabla 10: Comparación de las medidas de la velocidad del viento. Fuente: (El Autor).

PROTOTIPO IMPLEMENTADO	ESTACIÓN METEOROLÓGICA “VAISALA QML210”	DESVIACIÓN	ERROR (%)
0,63	0,36	0,27	0,76
2,21	2,52	-0,31	2,52
2,38	2,52	-0,14	-0,06
1,10	0,72	0,38	0,53
0,14	0,36	-0,22	-0,61
1,11	1,08	0,03	0,03
2,71	2,52	0,19	0,08
2,51	3,24	-0,73	-0,23
2,65	2,88	-0,23	-0,08
0,96	0,72	0,24	0,33
		ERROR	0,33

f.4 SENSOR DE DIRECCIÓN DEL VIENTO

En la tabla 11 se da a conocer el promedio de los datos de la dirección del viento en los horarios anteriormente descritos en el sensor de velocidad del viento.

Tabla 11: Comparación de las medidas de la dirección del viento. Fuente: (El Autor).

PROTOTIPO IMPLEMENTADO	ESTACIÓN METEOROLÓGICA “VAISALA QML210”	DESVIACIÓN	ERROR (%)
169,00	199,00	-30,00	-0,15
256,00	286,00	-30,00	-0,10
253,57	190,00	63,57	0,33
124,00	95,00	29,00	0,31
222,00	172,00	50,00	0,29
247,00	248,00	-1,00	0,00
314,00	321,00	-7,00	-0,02
215,00	259,00	-44,00	-0,17
172,00	209,00	-37,00	-0,18
232,00	146,00	86,00	0,59
		ERROR	0,09

f.5 SENSOR DE PRECIPITACIÓN

En la siguiente tabla se muestra la cantidad de agua caída en determinados intervalos de tiempo, en días que se produjo precipitación.

Tabla 12: Comparación de las medidas de precipitación. Fuente: (El Autor).

PROTOTIPO IMPLEMENTADO	ESTACIÓN METEOROLÓGICA "VAISALA QML210"	DESVIACIÓN	ERROR (%)
0,20	0,10	0,10	1,00
3,00	3,50	-0,50	-0,14
3,00	3,80	-0,80	-0,21
0,40	0,50	-0,10	-0,20
1,40	1,00	0,40	0,40
1,60	2,00	-0,40	-0,20
0,40	0,40	0,00	0,00
		ERROR	0,09

f.6 PRUEBAS DE ALMACENAMIENTO EN UNA MEMORIA SD

El datalogger realiza el almacenamiento de cada una de las variables, consta de una tarjeta SD con la capacidad de almacenamiento de 2GB; dentro de este parámetro de almacenamiento, la parte fundamental ha sido el poder almacenar los datos con su respectiva fecha y hora. En la tabla 14 se muestra el formato para guardar los datos en la SD.

Tabla 13: Registro de las variables del prototipo de estación meteorológica en una memoria externa. Fuente: (El Autor).

Temperatura	Humedad	Vel_del viento	Direc_del viento	Precipitación	Hora	Fecha
24.10	54.90	1.20	112	0.00	14:23	2014/9/12
24.10	53.90	2.40	113	0.00	14:23	2014/9/12
24.50	53.40	0.00	111	0.00	14:24	2014/9/12
24.50	53.40	1.20	111	0.00	14:24	2014/9/12
24.60	53.00	2.40	112	0.00	14:25	2014/9/12
24.80	52.90	1.20	338	0.00	14:25	2014/9/12
24.80	52.10	2.40	337	0.00	14:26	2014/9/12
24.90	52.00	1.20	340	0.00	14:26	2014/9/12

f.6.1 ESPECIFICACIONES DE ALMACENAMIENTO

- ✓ Capacidad de almacenamiento en la microSD de hasta 7240 archivos (capacidad por archivo de 266KB).
- ✓ Número de muestras almacenadas por día es de 2880 en periodos de 30 segundos.
- ✓ Datos almacenados en tiempo real de cada una de las variables meteorológicas.
- ✓ Capacidad de almacenamiento de 2GB teóricamente (1.83GB).
- ✓ Aproximadamente almacenará datos durante 19 años de manera continua.

f.7 PRUEBAS DE COMUNICACIÓN PARA REGISTRO DE LOS DATOS

El registro de datos de cada uno de los sensores se los hará en emomc.org donde se almacenará los sucesos o cambios producidos en cada uno de los distintos sensores implementados, en cualquier momento que se desee observar los datos se lo podrá hacer ingresando la dirección del servidor (www.emomc.org/VicentePoma) o la URL de la estación meteorológica (<http://unlemeteorologica.github.io/>). En la figura 53 se muestran las gráficas de las variables sensadas en la página del servidor emomc.org, mientras que en las figuras 54, 55, 56, 57, 58, 59 se muestra la introducción, presentación, sensores con sus respectivas gráficas de medición importadas desde el servidor antes mencionado.

UNIVERSIDAD NACIONAL DE LOJA

Área de la Energía, las Industrias y los Recursos Naturales No Renovables

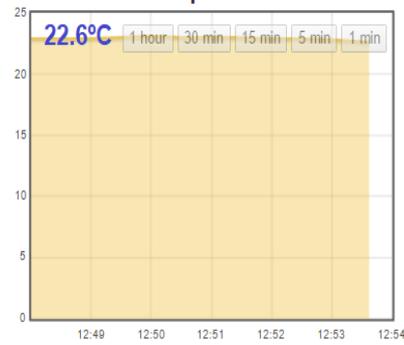
"DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEREOLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA."

Autor: Vicente Rigoberto Poma Godoy.

Director: Ing. Juan Pablo Cabrera Samaniego

Altitud: **2137msnm**
Latitud: **4.030**
Longitud: **79.200**

Temperatura



Humedad



Velocidad del Viento



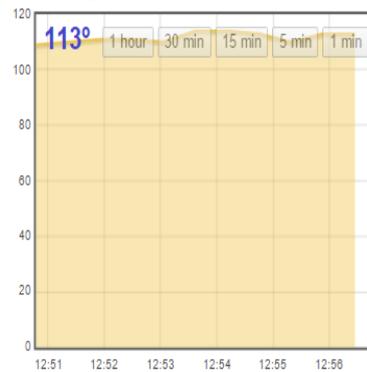
Dirección del Viento

N 337.5° - 22.5°
NE 22.5° - 67.5°
E 67.5° - 112.5°

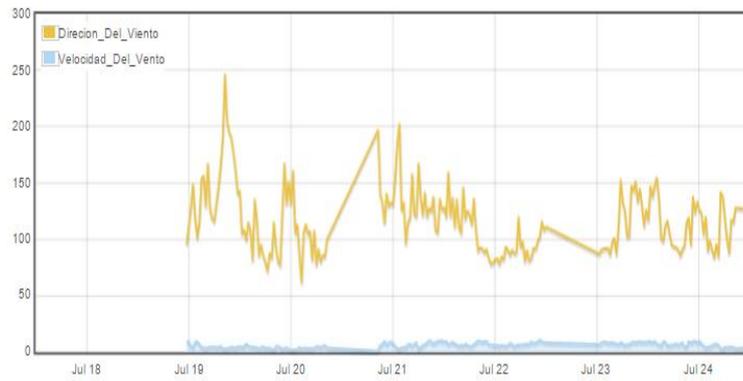
Velocidad del Viento



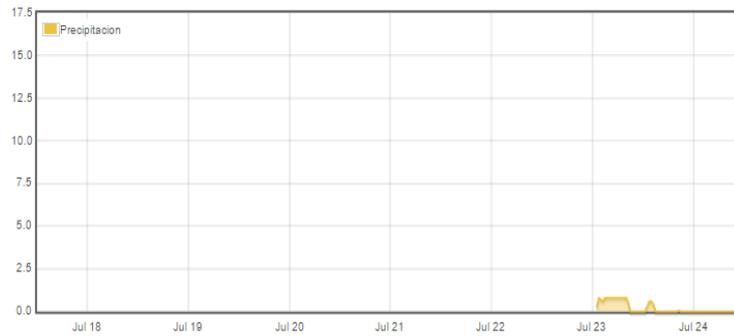
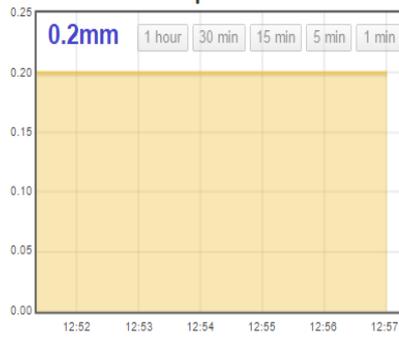
Dirección del Viento



SE 112.5 ° - 157.5 °
 S 157.5 ° - 202.5 °
 SW 202.5 ° - 247.5 °
 W 247.5 ° - 292.5 °
 NW 292.5 ° - 337.5 °



Precipitación



Powered by openenergymonitor.org | v8.3.5

Figura 53: Gráfica de los datos de cada una de las variables sensadas en el servidor emoncms.org. Fuente: (El Autor).



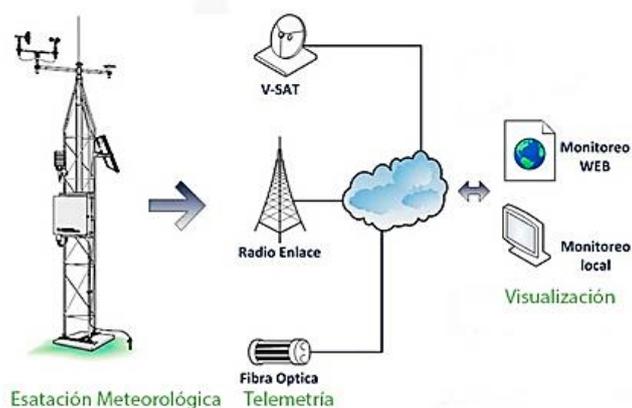
SISTEMAS DE MEDICIÓN CON TELEMETRÍA PARA ESTACIONES METEOROLÓGICAS

Los sistemas de telemetría se introdujeron a principios del siglo 20 para ser utilizados de supervisión en la naturaleza, controlar la distribución de energía eléctrica, entre otras. Hoy en día la telemetría es de vital importancia en diversas áreas, principalmente en la industrial, tecnológica y agropecuaria.

Los sistemas de monitoreo en la actualidad ofrecen importantes avances en lo que se refiere a la meteorología, dando confiabilidad en la transmisión correcta de los datos desde sitios remotos; así como también la visualización de los mismos en los centros de control o de monitoreo, para esto se desarrolla una interfaz visual que permitirá monitorear, registrar e integrar información a un determinado sistema, además se podrá acceder a la información desde cualquier parte del mundo.

La transmisión de los datos se los puede hacer por medios guiados como también no guiados, tomando en cuenta las condiciones favorables para poder implementar una red adecuada; de las cuales pueden ser: satelital, radio frecuencia, fibra óptica o ethernet.

Las estaciones meteorológicas en coexistencia con la tecnología hacen que se origine gran cantidad de avances en diferentes áreas de manera rápida y eficiente, como por ejemplo: en lo que respecta a generación y transmisión de energía limpia, recursos hídricos, producción industrial, producción del agro, turismo y recreación; por lo que es muy necesario contar con redes inteligentes que presten servicios complementarios para el desarrollo de proyectos.



02:34 PM 15

Calendario

December 2014

Do	Lu	Ma	Mi	Ju	Vi	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Figura 54: Gráfica de inicio en el diseño web del prototipo de estación meteorológica.
Fuente: (El Autor).

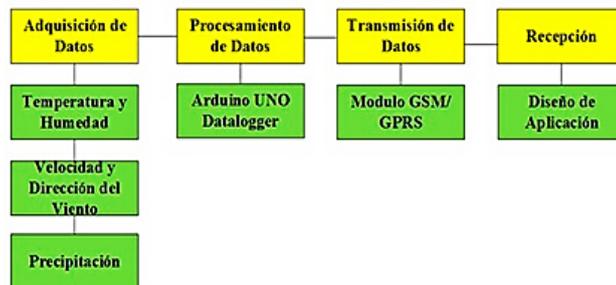


Prototipo de Estación Meteorológica

El desarrollo de éste proyecto de investigación está enfocado a medir variables climáticas; para alojarlas y presentarlas en la web, luego de un tratamiento adecuado de las señales adquiridas; logrando de esta manera cumplir con el objetivo de diseñar y construir un prototipo de estación meteorológica con monitoreo remoto.

El diseño del software versa en el desarrollo de un código entendible por el micro controlador de la placa arduino, el mismo que tiene sus distintas etapas en donde se puede calcular, calibrar, guardar los datos que obtienen los sensores en una microSD y enviar los mismos a través de una red celular para así llegar a su destino final como lo es el servidor emoncms.org.

Para la implementación del prototipo de estación meteorológica se detallan en cuatro partes principales y son:



Adquisición de datos.- Aquí se encuentran los sensores que estarán tomando las respectivas mediciones de temperatura, humedad, velocidad y dirección del viento y la precipitación.

Procesamiento de datos.- Se adecuarán mediante un algoritmo de programación todos los datos que se obtienen de los sensores climatológicos para su posterior almacenamiento, envío y presentación.

Transmisión.- Se ha empleado el módulo SIM900 GSM/GPRS para el envío de datos a través de una red celular haciendo uso del internet. Cabe mencionar que se enviará una trama con los datos de todos los sensores cada 20 segundos hacia el servidor.

Presentación de los datos.- Para la presentación se ha desarrollado una interfaz gráfica en el servidor gratuito y también en la página web publicada con la ayuda de GitHub, también cabe mencionar que el servidor brinda la posibilidad de exportar las gráficas de las variables climáticas a ésta página desarrollada donde se puede observar el estado meteorológico del sector.

02:35 PM

Calendario

Diciembre 2014						
Do	Lu	Ma	Mi	Ju	Vi	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Figura 55: Gráfica de proyecto en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).



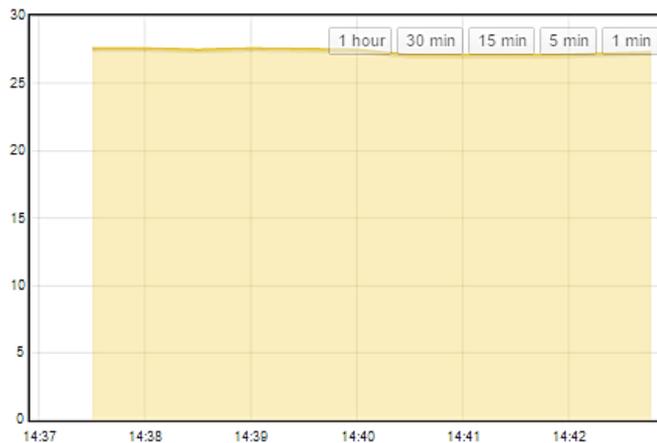
Termómetro e Higrómetro

El termómetro es un instrumento que mide las variaciones de la temperatura producidas por la velocidad a la que se mueven las moléculas de oxígeno y de nitrógeno. Su unidad de medida es en grados Celcius o Fahrenheit, y el rango de operación es -40 a 80 °C, con una precisión de $\pm 0.5^\circ\text{C}$.

También tenemos el higrómetro que será el encargado de medir la humedad relativa del aire, y su unidad de medida estará expresada en porcentaje (%), su rango de operación es de 0% a 100%, con una precisión de $\pm 2\%$. Para medir estas dos variables tenemos el sensor DHT22 que a continuación se presenta.



Gráfica de Temperatura (°C)



Gráfica de Humedad Relativa (%)

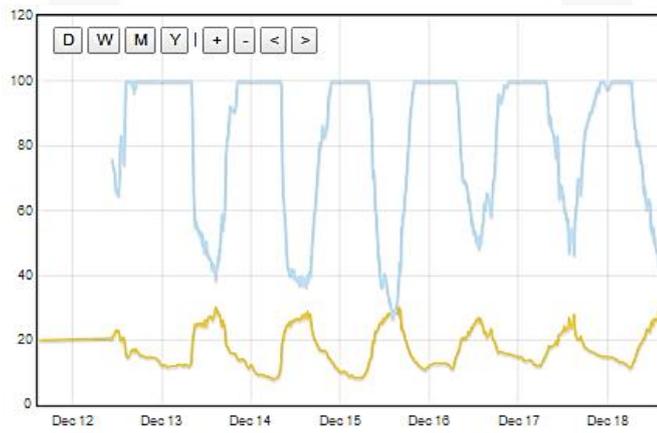
02:42 PM
37

Calendario

Diciembre 2014						
Do	Lu	Ma	Mi	Ju	Vi	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			



Gráfica comparativa de Temperatura (°C) y Humedad Relativa (%)



Designed by: Vicente Poma

viche_poma28@hotmail.es

Figura 56: Gráfica de temperatura en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).



Anemómetro y Veleta

Anemómetro, es el encargado de medir la velocidad de una masa de viento, ésta se puede expresar en kilómetros por hora (Km/h) o en metros por segundo (m/s), su rango de medición es de 1 a 322Km/h, con una precisión de ± 3 Km/H.

Veleta, es la encargada de determinar la dirección que proviene el viento, dándonos una posición cardinal, su unidad de medida esta expresada en grados sexagesimales, sus medidas están dentro de un rango de los 0° hasta los 360°, con una precisión de ± 3 °. Los sensores se encuentran implementados en un solo dispositivo de marca DAVIS modelo 6410 como se muestra en la figura.



03:06 PM 53

Calendario

Diciembre 2014						
Do	Lu	Ma	Mi	Ju	Vi	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

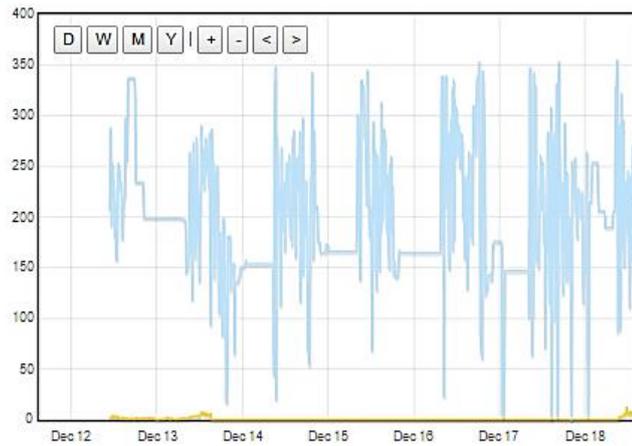
Gráfica de Velocidad del Viento (Km/h)



Gráfica de Dirección del Viento y su interpretación (°)

N	337.5°	- 0°	- 22.5°	NE	22.5°	- 45°	- 67.5°
E	67.5°	- 90°	- 112.5°	SE	112.5°	- 135°	- 157.5°
N	157.5°	- 180°	- 202.5°	NE	202.5°	- 225°	- 247.5°
E	247.5°	- 270°	- 292.5°	SE	292.5°	- 315°	- 337.5°

Gráfica comparativa de Velocidad (Km/h) y Dirección del Viento (°)



Designed by: Vicente Poma

viche_poma28@hotmail.es

Figura 57: Gráfica de velocidad y dirección del viento en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).



Universidad Nacional de Loja
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

INICIO
PROYECTO
TEMPERATURA
VEL_DIR_VIENT
PRECIPITACIÓN
GALERÍA



ESTACIÓN METEOROLÓGICA

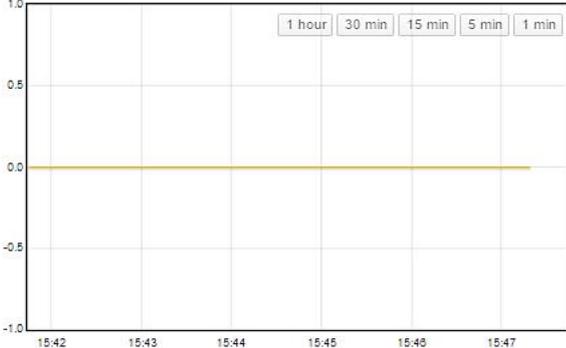


Pluviómetro

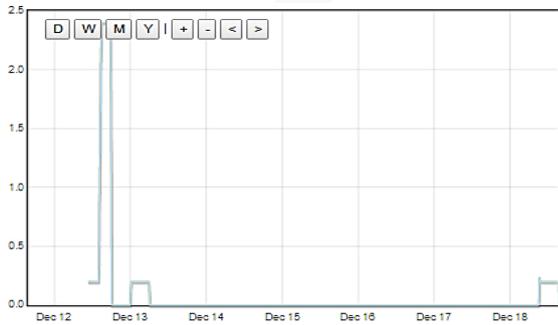
El pluviómetro es un instrumento que se emplea en las estaciones meteorológicas para recoger y medir la cantidad de agua caída. Su unidad de medida se expresa en milímetros (mm) de altura y su diseño básico consiste en una abertura superior de tipo embudo por donde el agua se lleva hasta penetrar por un agujero con un diámetro de 2mm cayendo en un balancín donde se almacena una cierta cantidad que a posteriori produce un vuelco, generando así una variación que servirá para interpretar su medida y determinar la cantidad de agua caída. El dispositivo es de marca DAVIS modelo 7852, su rango de operación es de 0 a 999.9mm, con una precisión de $\pm 3\%$.



Gráfica de Cantidad de Precipitación (mm)



Gráfica de Cantidad de Precipitación en distintos periodos de tiempo (mm)



03:47^{PM}₄₅

Calendario

Diciembre 2014						
Do	Lu	Ma	Ju	Vi	Sa	
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Designed by: Vicente Poma
viche_poma28@hotmail.es

Figura 58: Gráfica de precipitación pluvial en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).



Universidad Nacional de Loja
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

INICIO
PROYECTO
TEMPERATURA
VEL_DIR_VIENT
PRECIPITACIÓN
GALERÍA



ESTACIÓN METEOROLÓGICA

Galería

Vídeo del prototipo de estación meteorológica con sistema de telemetría implementado.

Prototipo de Estación Meteorológica



ARDUINO UNO



SIM900 GSM/GPRS

Vídeo de prueba para la implementación del prototipo de estación meteorológica con sistema de telemetría.

LISTA DE REPRODUCCIÓN | 2 / 2 Estacion Meteorologica



Los dispositivos que se emplearon para la implementación del prototipo se los dará conocer en las siguientes gráficas; primeramente está la placa Arduino UNO que fue fundamental para el procesamiento digital de los datos, seguida del módulo SIM900 GSM/GPRS, el mismo que se comunica con la placa Arduino y con el servidor emoncms.org, transmitiendo de esta forma las variables climáticas del proyecto desarrollado.




Los sensores que se emplearán para la adquisición de los datos son: DHT22, el Anemómetro, la Veleta y el Pluviómetro. Para el almacenamiento de los datos tenemos el Módulo SD de lectura y escritura con su respectiva Memoria microSD de 2GB de capacidad.








02:39

PM 99

Calendario

Diciembre 2014

Do	Lu	Ma	Mi	Ju	Vi	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Designed by: Vicente Poma
viche_poma28@hotmail.es

Figura 59: Gráfica de galería en el diseño web del prototipo de estación meteorológica. Fuente: (El Autor).

f.8 CARACTERÍSTICAS TÉCNICAS DEL PROTOTIPO DE ESTACIÓN METEOROLÓGICA

El prototipo de estación meteorológica tiene las siguientes características:

Tabla 14: Características del prototipo de estación meteorológica. Fuente: (El Autor).

PROTOTIPO DE ESTACIÓN METEOROLÓGICA	
Voltaje de alimentación	6V
Corriente para pines E/S	40 mA
Memoria Flash del procesador	32 KB
RAM del procesador	2KB
EEPROM del procesador	1 KB
Rango de medición de Temperatura	-40 a + 80 °C
Precisión de temperatura	± 0.5 Celsius
Resolución de temperatura	0,1 °C
Rango de medición de la humedad	0 a 100%
Precisión de humedad	± 2% (Max 5% RH)
Resolución humedad	0,1% de HR
Rango de medición de la velocidad del viento (anemómetro)	1 a 322 km/h
Precisión de la velocidad del viento	±3 km/h o ±5%
Rango de medición de la dirección del viento (veleta)	0° - 360°
Precisión de la dirección del viento	±3°
Periodo de muestreo del anemómetro y veleta	2.25 segundos
Rango de medición de precipitación diaria	0.0 mm a 999.8 mm
Precisión de la precipitación	± 3%, ± un conteo entre 0.2 y 50 mm/h ± 5%, ± un conteo entre 50 y 100 mm/h
Resolución de precipitación	0.2 mm
Periodo de almacenamiento de cada muestra de datos	30 segundos
Temperatura de operación de la estación meteorológica.	-40° a +65°C
Módulo GSM/GPRS	Voltaje de alimentación 5V Consumo de energía 1A Quad-Band 850/900/1800/1900 MHz. GPRS clase 10/8 multi-slot Clase de estación móvil B Velocidad de transmisión 115kbps
Servidor prediseñado	www.emoncms.org Periodo de actualización 30 segundos
Página Web	http://unlemeteorologica.github.io

f.9 ANÁLISIS ECONÓMICO Y COMPARATIVO DEL PROTOTIPO DE ESTACIÓN METEOROLÓGICA DISEÑADA, CON UNA COMERCIAL

El análisis económico del prototipo de estación meteorológica implementado, se realiza en base a la cantidad de dispositivos y al precio que se encuentran cada uno en el mercado actual, de esta manera obtener el valor aproximado del prototipo implementado para una posterior comparación económica con estaciones meteorológicas, cabe recalcar que el proyecto fue financiado única y totalmente por el autor. Además se hace un análisis comparativo de las características importantes del prototipo implementado con otras estaciones meteorológicas.

f.9.1 COSTO DEL DISPOSITIVO IMPLEMENTADO

En la tabla 15 se detalla los materiales utilizados, con su respectivo precio, los mismos que dan un costo total de \$ 550.

Tabla 15: Presupuesto del hardware para la estación meteorológica. Fuente: (El Autor).

DESCRIPCIÓN	CANT.	COSTO EN DÓLARES
Arduino UNO.	1	\$ 42,00
Sim900 GSM/GPRS.	1	\$ 80,00
Sensor de temperatura y humedad relativa.	1	\$ 20,00
Sensor Pluviómetro.	1	\$ 80,00
Sensor de velocidad y dirección del viento.	1	\$ 175,00
Placa del reloj	1	\$ 3,00
Módulo para memoria SD	1	\$ 15,00
Memoria SD	1	\$ 10,00
Caja hermética	1	\$ 15,00
Protección para sensor DHT22	1	\$ 25,00
Tarjeta de RTC, SD y Sensores	1	\$ 15,00
Otros	1	\$ 70,00
TOTAL		\$550,00

f.9.2 COMPARACIÓN DEL PROTOTIPO IMPLEMENTADO CON ESTACIONES METEOROLÓGICAS COMERCIALES

La comparación entre dos o más sistemas es de gran importancia a la hora de considerar lo que se desea implementar, en el caso de las estaciones meteorológicas se ha considerado que tengan los sensores que posee el prototipo de estación meteorológica para la medición de variables climáticas, la precisión de los dispositivos es un factor muy importante, misma que puede variar por marca, modelo y nivel de precisión del dispositivo, cabe mencionar también que los precios variarán de acuerdo al nivel de precisión con el que mide un sensor una determinada variable, y de acuerdo a estas características los vendedores los clasifican a las estaciones como profesionales y no profesionales, en este caso en particular se empleó tres sensores de marca DAVIS que es bastante reconocida en el mercado por la fabricación de estos dispositivos y venta de estaciones meteorológicas de buena calidad.

En la tabla 16 se muestran tres sistemas de meteorología, el prototipo implementado, un americano y un finlandés, donde se muestran algunas de sus características y los dispositivos con los que cuenta dentro de la factura de compra.

***Tabla 16:** Comparación del prototipo diseñado con las estaciones meteorológicas comerciales. Fuente: (El Autor).*

CUADRO COMPARATIVO DE ESTACIONES METEOROLÓGICAS

Parámetro	Estación 1	Estación 2	Estación 3
Imagen			
Marca	-----	Davis Instruments	VAISALA
Modelo	-----	Volataje value 2 6152	QML210
Precio	\$ 550.00	\$ 702.60 (España)	\$ 8,500.00 (Ecuador)
Voltaje de operación.	6V	6V	12 V
Batería	No	Si	Si
Módulo solar	No	Si	Si
Tecnología para la transmisión de datos.	GSM/GPRS	Wireless (Transmisor y Receptor)	GSM/GPRS WAVECOM
Tipo de comunicación.	Unidireccional	Unidireccional	Unidireccional
Muestra reloj (RTC).	Si	Si	Si
Interfaz utilizada para la lectura de datos.	Interfaz en emoncms.org server. Página web subida con GitHub. Almacena en SD cada 30 seg.	Consola con software propio.	Servidor del proveedor. Con creación de usuarios. Almacena en SD cada 60 minutos

Intervalo de envío de datos.	Cada 20 segundos al servidor. El servidor actualiza en el rango de 30 segundos.	Varían de acuerdo a los sensores para lo cual se toman en cuenta las características de estos y el software.	Cada hora al servidor web.
Sensores integrados.	Sensores de temperatura y humedad modelo DHT22; anemómetro y veleta Marca Davis, Modelo 6410; pluviómetro Marca Davis, Modelo 7852M.	Sensores de temperatura y humedad Marca Davis, Modelo 6382; anemómetro y veleta Marca Davis, Modelo 6332; pluviómetro Marca Davis, Modelo 6152C.	Sensores de temperatura Marca Vaisala, Modelo HMP155; anemómetro y veleta marca YOUNG, Modelo 5103; pluviómetro Marca Texas Electronics, Modelo TR-625M.
Pantalla de visualización.	No	Si	No
Breve descripción de las estaciones meteorológicas.	Su puerto de comunicación es RS-232 con velocidades de transmisión configurable, acceso para modificar en pro a mejorar su sistema, sus registros de datos en el servidor pueden ser observados en diferentes intervalos de tiempo, muestra su historial completo desde que se empezó su transmisión, además son almacenados sus datos en una SD cada 30 segundos. Sus datos pueden ser observados en cualquier parte del país o del mundo, por lo que no se limita a ciertas distancias. Tiene la opción de incrementar más sensores que se crea conveniente.	Posee una consola con un sistema propio que permite ver los datos, historiales, establecer y borrar alarmas, cambiar los modelos de estaciones, introducir números de calibración, configurar y ver gráficos, seleccionar sensores y leer el pronóstico meteorológico, su puerto de comunicación puede ser RS-232 o USB a una velocidad de 19200 Baud. Para la transmisión de los datos lo hace mediante radio con un alcance de 300m sin obstáculos en línea de vista y de 12 a 90m con obstáculos, entre el transmisor y el receptor.	El sistema básico ofrece, puertos RS-485 y SDI-12 RS-232 para la conexión con casi cualquier tipo de telemetría, terminales y sensores inteligentes. Además dentro de los materiales para instalación incluye: Mástil de 10 metros galvanizado. Regulador de voltaje. Pararrayos y varilla de puesta a tierra. Sensor de radiación solar. Sensor de radiación solar reflejada. Sensor de presión barométrica.
Hecho en:	Ecuador	Estados Unidos	Finlandia

g DISCUSIÓN

Para el desarrollo del diseño e implementación de un prototipo de estación meteorológica, con un sistema de telemetría en primera instancia se consideró los dispositivos que se deseaba implementar para medir los principales factores climáticos en la ciudad de Loja sector de la Argelia, con el objetivo que se tome como un estudio de utilidad para el desarrollo de otros proyectos o el mejoramiento del mismo, cabe mencionar que las variables a medir son: temperatura, humedad relativa, velocidad del viento, dirección del viento y precipitación.

Para obtener la medida de la temperatura y la humedad existe en el mercado un sensor de buena calidad con prestaciones bastante fiables y un coste relativamente económico, el sensor utilizado es el DHT22 con características de transmisión de hasta 20 metros, un rango de medición de -40 a $+80$ °C en lo que se refiere a la temperatura y en cuanto a la humedad relativa su rango es 0 a 100%; con una precisión de temperatura ± 0.5 grados Celsius y de Humedad relativa $\pm 2\%$ (Max 5% RH) en teoría; las mediciones que se realizaron con este sensor son bastante fiables con respecto a la temperatura, las mismas que son presentadas en el servidor web (emonmcs.org) y guardadas en una SD en un intervalo de tiempo de 20 y 30 segundos respectivamente, por lo que se considera que son mediciones en tiempo real, mientras que las mediciones de Instituto Nacional de Meteorología e Hidrología (INAMHI) con respecto a la ciudad de Loja, son colgadas en su página web en intervalos de tres horas empezando desde las 07:00am hasta las 19:00pm con las cuales se realizó las comparaciones respectivas; por ejemplo una muestra tomada a las 07:00am del día 16 de octubre del 2014 el prototipo midió una temperatura de 14°C mientras que la medida del INAMHI colgada en la página a esa hora fue 13°C y otra muestra tomada al siguiente día a las 19:00pm el prototipo presenta una medida de 15.30°C siendo inferior a la medida presentada por el INAMHI que es de 17°C lo que nos da un nivel de error promedio bastante bajo (0,05%) como se muestra en la Tabla 9 cabe indicar que las muestras fueron tomadas en dos días consecutivos.

La humedad relativa es un factor importante aunque en estaciones de la localidad no ha sido tomada en cuenta para su respectiva medición por lo que no se puede realizar comparación alguna y únicamente se muestran las medidas obtenidas en la Tabla 10. Las mismas que tienen coherencia con respecto a la temperatura suscitada en esos días.

El anemómetro empleado para obtener la velocidad del viento es de Marca DAVIS modelo 6410, con muy buenas características de sensado y trabaja con un Reed Switch que hace el conteo de flancos cada vez que se complete una vuelta de las copas, para su respectivo cálculo se toma en cuenta el valor obtenido en un intervalo de tiempo, lo multiplica por una constante y se divide para dicho intervalo de tiempo en el que se produjeron cierta cantidad de vueltas de las copas del sensor, de esta manera se obtiene la velocidad en millas por hora, para transformar esta medida a kilómetros por hora (Km/h) se realiza el cálculo en base al número de vueltas que equivalen a una milla por hora que se muestra en el anexo correspondiente al sensor; en cuanto a la comparación de las mediciones obtenidas por el prototipo se las debió adecuar para compararlas con la estación meteorológica de marca Vaisala modelo QML210 ya que esta estación muestra las medidas máximas, mínimas y promedio en el lapso de una hora, el sensor que utiliza esta estación es de la marca Young, el intervalo de tiempo para el almacenamiento de los datos en la SD del prototipo es de 30 segundos, por lo que se procedió a obtener el promedio de las medidas de 07:00am a 08:00am y así para el resto de medidas tomadas, solo por considerar el desfase entre las dos estaciones presentamos dos datos del día 9 y 10 de octubre del 2014 tomadas en el lapso de 07:00am a 08:00am, la medida del prototipo fue de 0,63 Km/h y en la estación Vaisala es de 0,36 Km/h, como se muestra en la Tabla 11, en base a las distintas medidas tomadas para el cálculo del error nos da un porcentaje de 0,33 % siendo un nivel de error favorable para el prototipo diseñado.

La dirección del viento se obtiene por la veleta de marca Davis modelo 6410 que de hecho viene en conjunto con el anemómetro pero realizan sus funciones independientemente, este sensor básicamente está compuesto por un potenciómetro de valor de 20k Ω , el mismo que para su calibración necesita de un circuito divisor de voltaje o pull up (Figura 27) por lo que es necesario hacer una conversión analógico digital(ADC) para establecer un rango de 0 a 359 que corresponden a todos los puntos cardinales; dentro del circuito tenemos un pulsador que al presionarlo automáticamente el procesador ATMEGA328 leerá la posición de la veleta con un valor de 0 ubicándole como la posición norte, el procesador guardará este dato para poder obtener las demás direcciones en base a esta referencia, para el cálculo del resto de direcciones se toma en cuenta las 16 posiciones cada una de 22,5° que se muestran en la hoja de datos en la sección de anexos, que darán un total de 360° conociendo todos estos datos, tenemos la opción de poder determinar los rangos

para la lectura de la posición actual a la que se encuentra fluyendo el viento. En la figura 44 se muestra la gráfica y los grados del que proviene el viento, para este proyecto se consideró tomar rangos de 45° para las ocho posiciones que se presentan como son: Norte, Noreste, Este, Sureste, Sur, Suroeste, Oeste, Noroeste. Las medidas tomadas en el prototipo son comparadas igualmente con la estación Vaisala que utiliza un sensor de marca Young, considerando las mismas horas en que se tomaron las muestras para la velocidad del viento, por ejemplo el día 9 de octubre del 2014 se tomó de 07:00am a 08:00am la medida del prototipo fue de 169° y en la estación Vaisala es de 199° , mientras que al siguiente día tomando la hora de promedio de 19:00pm a 20:00pm el prototipo presentó la medición de 232° y la estación comercial 146° , en la Tabla 12 se presentan las mediciones realizadas que dan un porcentaje de error de 0,09%, el mismo que es favorable respecto a la precisión del prototipo sobre esta variable sensada.

La intensidad de lluvia caída es medida en el prototipo de estación meteorológica por el sensor pluviómetro de marca Davis que cuenta con un Reed Switch bastante parecido al del anemómetro ya que su conteo lo realiza por cambio de flancos (Interrupciones), está compuesto por un balancín con dos copas que al contener cierta cantidad de agua producirán un vuelco dando así un cambio de flanco. La cantidad de agua puede ser regulada de acuerdo a la hoja de datos del sensor, en el caso del diseño la cantidad es de 0.20 mm; para el cálculo de esta variable se toma en consideración la medida anteriormente descrita multiplicada por el vuelco y posteriormente se realiza la sumatoria de los vuelcos, además presenta los datos después de cada vuelco producido; en un lapso de 6 horas se lo configuró para que se resetee y vuelva a comenzar su conteo; para obtener el nivel de error se lo realizó con el sensor TR-525M de marca Texas Electronics de la estación Vaisala; también se tomó en cuenta dos factores importantes como son: primero la cantidad de agua almacenada por hora que es lo que presenta la estación Vaisala, segundo el tiempo en que se resetea el sensor del prototipo implementado y tercero las medidas fueron tomadas en los días que se produjo precipitación pluvial; el segundo se consideró como más importante ya que al resetearse dentro de una hora de cálculo podría producir un nivel de error alto; los datos que se emiten a continuación son producto de los días en que se produjo precipitación pluvial por ejemplo la medida del prototipo fue de 0.20 mm y en la estación Vaisala es de 0.10 mm, mientras que otra medición tomada por el prototipo es 0.40 mm y la estación comercial 0.40 mm, en la Tabla 13 se

especifican los valores de las precipitaciones con las que se obtuvo el nivel de error de 0,09 %.

Una de las características de las estaciones meteorológicas es de contar con un respaldo de la información que generan cada una de las variables medidas en el tiempo por lo que se implementó un datalogger con la ayuda de un módulo SD que se acopla a la placa Arduino, donde almacenará las variables cada 30 segundos en una memoria microSD. Para esto se determinó que se necesita de un RTC para saber el tiempo en que se produjo una medición el cual se desarrolló en base al circuito integrado DS1307, primeramente se crea un archivo de texto plano o de extensión .txt, luego el formato en el que se escribirá el nombre de las variables, acto seguido procede a guardar el valor de cada variable con la hora y fecha respectiva tal como se muestra en la Tabla 14.

Finalmente, se describe el envío de los datos al servidor prediseñado como es emoncms.org, para lo cual se debe crear una cuenta que le dé acceso para recibir datos; al tener ya un registro en este servidor se genera un apikey que se lo configurará en el procesador ATMEGA328, entonces para el transporte de los datos a través de una red celular se requiere de un módulo SIM900 GSM/GPRS el cual recibe las instrucciones del procesador para establecer comunicación con el servidor y posterior envío de la trama de datos en donde se empaquetan todas las variables sensadas. Para este proceso primeramente el módulo debe responder a las siguientes instrucciones como: reporte de calidad de señal una vez que se encuentra enganchado a la red, ajustar al módulo para aplicaciones IP, configurar el punto de acceso de la red, comprueba el estado del registro y posteriormente inicia la petición al protocolo HTTP, donde se configura la dirección URL del servidor con su respectivo apikey que le permitirá sincronizarse con el servidor para enviar los datos en intervalos de tiempo de 20 segundos, una vez realizado este proceso se finaliza la conexión. La cuenta creada en el servidor se la puede personalizar de forma adecuada para poder observar los valores de las distintas variables (ver figura 52) que se está recibiendo tal como se explica en la sección e.1.11.2, también se desarrolló un sitio web donde se encuentra personalizado con información sobre el prototipo de estación meteorológica con sistema de telemetría (ver figuras desde 53 a 58), las gráficas de las variables generadas en el servidor se las exportó para publicarlas en este sitio web ya que es una de las bondades que presta emoncms.org.

h CONCLUSIONES

- ✓ El prototipo implementado cumple con las principales funciones de una estación meteorológica al realizar mediciones de temperatura, humedad, dirección y velocidad del viento y la precipitación, además presenta varias ventajas sobre las comerciales siendo estas tanto económicas, de software, y desarrollo de conocimientos.
- ✓ El sistema de telemetría desarrollado para el envío de los datos al servidor prediseñado se lo realiza de manera gratuita por lo que se limita a normas y condiciones impuestas por el propietario, pero cumple a cabalidad con el objetivo planteado al presentar los datos tanto en la página del servidor empleado como el desarrollo web que se elaboró e implemento.
- ✓ Es preciso mencionar que la transmisión de los datos está sujeto a las condiciones de trabajo de la operadora celular que se empleó para realizar el envío de los mismos.
- ✓ Cabe mencionar que la comparación de los datos de temperatura entre la estación del INAMHI y el prototipo implementado fueron tomados en distintos intervalos de tiempo de acuerdo a las horas en que suben los datos a la página del INAMHI, mientras que para las comparaciones de los datos de dirección y velocidad del viento se tomó en cuenta el promedio de cada hora de estas variables ya que la estación VAISALA QML210, almacena y presenta en su página web las cantidades máximas, mínimas y promedios; para la precipitación se tomó en cuenta solo en los días que se produjo lluvias, se comparó la cantidad que se almacena durante una hora, debido a las características de medición de la estación VAISALA con la que se realizó las respectivas comparaciones.
- ✓ Los sensores implementados cumplen con los requerimientos técnicos como: temperatura de funcionamiento dentro de un rango de -40°C a $+65^{\circ}\text{C}$; rango de medición de acuerdo a cada sensor, el termómetro de -40°C a $+80^{\circ}\text{C}$ con un nivel de error de $\pm 0.5^{\circ}\text{C}$, el higrómetro tiene un rango de medición de 0% a 100% con

un nivel de error de $\pm 2\%$, la veleta de 0° a 360° con un nivel de error de $\pm 3^\circ$, el anemómetro de 0 a 322 Km/h con un nivel de error de $\pm 5\%$ y en pluviómetro de 0 a 999.8 mm con un nivel de error de $\pm 3\%$ a $\pm 5\%$; factores que fueron fundamentales en el proceso de selección y posterior construcción, lo que determinó que durante el análisis comparativo estos sensores presenten un nivel de error menor al 1% con respecto a las estaciones de referencia.

- ✓ Para el estudio del presente proyecto se utilizó plataforma libre como es Arduino, ya que posee características bastante eficientes, por tal razón se hizo uso del módulo el mismo que realizará las funciones como: obtener, procesar, registrar y enviar la orden al módulo GSM/GPRS para la transmisión de los datos de las variables climáticas planteadas en el respectivo proyecto.
- ✓ La estación meteorológica aparte de transmitir los datos al servidor prediseñado a través de una red celular; está en la capacidad de almacenarlos en una memoria microSD a manera de respaldos por cualquier necesidad a futuro o posible eventualidad en la transmisión, el formato de archivo es: temperatura, humedad, velocidad de viento, dirección del viento, precipitación, hora y fecha, las muestras tomadas tendrán intervalos de almacenamiento de 30 segundos configurados por software, creando un archivo por día de 266 KB ya que diariamente almacenará 2880 muestras; la capacidad total de almacenamiento de microSD es de 2GB lo que nos da un total de 7,240 días que se puede almacenar en dicha memoria.
- ✓ Para el acceso a la cuenta del servidor prediseñado se requiere un usuario y contraseña y así hacer uso de los beneficios que brinda esta cuenta gratuita; además posee algunas ventajas como: configurar el tiempo de actualización de las variables en el servidor, mostrar las gráficas de las variables climáticas en su ventana de Dashboard en donde se puede personalizar tanto en tamaño, ubicación y cantidad de variables a presentar en las mismas, muestra de forma numérica el valor de sus variables gracias a los feedvalue también nos permite obtener el código html para poder exportar la gráfica a una página web específica, entre otras ventajas.

i RECOMENDACIONES

Para futuras implementaciones en este trabajo se podría realizar algunos incrementos que a continuación se detallan:

- ✓ Desarrollar un servidor web con usuarios personalizados y un nombre de dominio para que trabaje dentro de una red de internet de manera independiente, tener acceso desde cualquier lugar donde se desee ingresar y conocer el estado climatológico en el sector donde se implementó el prototipo de estación meteorológica, lo que generará un costo adicional para la compra de una IP pública y posible servicio de hosting.
- ✓ Añadir sensores como: radiación solar, humedad de la tierra, entre otros que se crea conveniente, tomando en cuenta que este incremento requiere ampliación de código para el funcionamiento de cada uno de estos y de esta manera se puedan integrar al prototipo implementado.
- ✓ Implementar un sistema de alimentación autónomo basado en paneles solares, con un adecuado cálculo del consumo de energía supliendo de manera óptima la alimentación de cada uno de los dispositivos.
- ✓ Se recomienda ampliar la memoria del procesador (Arduino), debido a la poca cantidad de memoria que presta el actual (32KB), si fuera el caso que se realice posteriores implementaciones de sensores y por ende de código, este requerirá de mayor capacidad para su óptimo funcionamiento.
- ✓ Tomar en cuenta el voltaje que alimenta al Arduino para no sobredimensionarlo y provocar recalentamiento, por ende reinicio del dispositivo provocando fallas en el prototipo.
- ✓ Tener en cuenta de no sobrecargar las salidas de voltaje del Arduino al conectarle dispositivos adicionales, ya que estos no funcionarían de manera adecuada.

- ✓ El módulo SIM900 a pesar de tener la opción de encendido por software para su funcionamiento, es poco útil cuando se trata de transmisión de datos por GPRS ya que se debe activar la comunicación GPRS y Serial en el algoritmo del programa, lo que hace que se produzca conflicto entre estas cuando se implementa el algoritmo de encendido, por lo que no realiza la comunicación al servidor, pero si realiza las otras funciones como lectura y almacenamiento de las variables climáticas; entonces es necesario recomendar encender el módulo SIM900 manualmente y a continuación realizar un reset al Arduino para que se reinicien todos los procesos normales de funcionamiento (procesamiento, almacenamiento y transmisión de los datos).

- ✓ Al implementar una estación meteorológica es muy importante tener en cuenta la norma de seguridad de la caja en donde se alojarán el procesador y demás implementos, ya que estarán expuestos diversos factores como: calor, lluvia, polvo, que podrán producir daños en los dispositivos, por lo que es recomendable tener en cuenta la norma IP65.

- ✓ Tener en cuenta las futuras aplicaciones en el desarrollo de nuevos proyectos basados en el prototipo de estación meteorológica con sistema de telemetría o también en la implementación de nuevos sensores con el afán de presentar nuevos servicios meteorológicos para áreas importantes como la turística, agrícola, industrial y tecnológica.

j **BIBLIOGRAFÍA**

- [1]. Fundación Española para la Ciencia y la Tecnología. Meteorología y Climatología. España : Villena Artes Gráficas, 2004.
- [2]. ABAD, Walberto. Diseño e implementación de un prototipo de estación meteorológica autónoma para el IASA I (ESPE). Quito : s.n., 2006. Vol. 1, 1.
- [3]. PALMA, Victor; RAMIREZ, Francisco. Estación Meteorológica multiparamétrica sincronizada con GPS y monitorizada a través de internet . El salvador : Universidad de El Salvador , 2013. 1.
- [4]. Organización Meteorológica Mundial .[En línea] 2010. [Citado el: 17 de 04 de 201.] https://www.wmo.int/pages/prog/www/ois/Operational_Information/Publications/WMO_386/WMO_386_Vol_I_2009_es.pdf.
- [5]. VILLALTA, Higinio; PERDOMO, Godofredo. Implementación de una estación meteorológica. San Salvador : Universidad de el Salvador , 2013. 1.
- [6]. MIRANDA, Roberto y POZO, Pablo. Diseño y pruebas de la red de transmisión de datos. Quito : Escuela Politécnica Nacional, 2012.
- [7]. Google.com. Telefonía móvil celular, GSM. [En línea] [Citado el: 31 de 05 de 2014.] <https://www.google.com.ec/#q=Telefon%C3%ADa+m%C3%B3vil+celular%2C+GSM+pdf>.
- [8]. Imágenes de sistemas celulares. [En línea] [Citado el: 01 de 06 de 2014.] www.google.com.ec/search?q=red+celular+de+morelos&source=lnms&tbm=isch&sa=X&ei=6n6MU4GLLum0sQS4qoCQBg&ved=0CAYQ_AUoAQ&biw=1137&bih=741&dpr=0.9#q=sistema+celular&tbm=isch&facrc=_&imgdii=_&imgcr=7UmZaiXolWxFrM%253A%3BfuvveTvLVIP9CM%3Bhttp%253A%252F%252F.
- [9]. VELASCO, Nicolas. Sistema Embebido para la conexión de un PLC Siemens S7-200 a la red GSM. Sevilla : Universidad de Sevilla, Noviembre, 2005 . Vol. 1.
- [10]. PRIETO, Francisco. Transmisión de imágenes de vídeo mediante Servicios Web XML sobre J2ME. Sevilla : Universidad De Sevilla, 2007.
- [11]. SÁNCHEZ, Juan Andrés. Análisis y Estudio de Redes GPRS. Valdivia : Universidad Austral de Chile, 2005. Vol. 1.
- [12]. BERNAL, Ivan. Comunicaciones Inalámbricas GPRS. Quito : Escuela Politécnica Nacional, 2007.

- [13]. LOAIZA, José. Diseño e implementación de un sistema de medición de energía eléctrica inteligente para el uso doméstico. Loja : Universidad Nacional de Loja, 2014. 1.
- [14]. VEGA, Maria. Implementación de un sistema de publicidad visual sobre tecnología celular GSM/GPRS. Quito : Escuela Politécnica Nacional, 2010.
- [15]. MOYANO, Jonathan. Telecontrol GSM. Telecontrol GSM - Desarrollo de proyectos. [En línea] [Citado el: 04 de 11 de 2014.] www.ucontrol.com.ar/forosmf/proyectos-con-pic/telecontrol-gsm-desarrollo-de-proyectos/5/?wap2.
- [16]. SCOTT, Chacon. Pro Git. California : Creative Commons, 2009. 1 Edición.
- [17]. R3, Arduino Mega2560. Arduino. [En línea] 2014. [Citado el: 15 de 05 de 14.] <http://arduino.cc/en/Main/arduinoBoardMega2560>.
- [18]. UNO, Arduino. Arduino. [En línea] 2014. [Citado el: 15 de 05 de 14.] <http://arduino.cc/en/Main/arduinoBoardUno>.
- [19]. Getech. SIM 900 GSM/GPRS. [En línea] 23 de 12 de 2013. [Citado el: 17 de 05 de 2014.] http://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield.
- [20]. SIMCom. <http://narobo.com>. <http://narobo.com>. [En línea] 19 de 08 de 2010. [Citado el: 14 de 10 de 2014.] [http://narobo.com/products/DroneCell/datasheet/HD .pdf](http://narobo.com/products/DroneCell/datasheet/HD.pdf).
- [21]. MGSYSTEM SOLUCIONES INFORMÁTICAS . Mercado Libre. [En línea] [Citado el: 24 de 05 de 2014.] http://articulo.mercadolibre.com.ec/MEC-40493862-6-modulo-sd-lectura-y-escritura-ideal-para-arduino-pic-_JM.
- [22]. CORTÉZ, Jennifer. Diseño de un sistema de control de temperatura y humedad para la conservación de puros habanos mediante microcontroladores. Proyecto Final de Carrera. Barcelona - España : Universidad Politécnica de Catalunya, 2012. 1.
- [23]. Openenergymonitor.org. [En línea] [Citado el: 10 de 07 de 2014.] www.emoncms.org/site/docs/inputsandfeeds.
- [24]. MeteoStar. MeteoStar. [En línea] Desarrollos Meteorológicos . [Citado el: 26 de 04 de 2014.] www.meteostar.com.ar/imagenes/contenido/estacionp.jpg.
- [25]. Amazon. Davis anemómetro Replacement Part # 6410. [En línea] DavisNet, 2014. [Citado el: 27 de 04 de 2014.] [http://www.amazon.com/Davis-Replacement-Anemometer-Part-6410/dp/B000WBR1QS/ref=sr_1_1?ie=UTF8&qid=1398940677 &sr=8-1&keywords=anemometer+davis](http://www.amazon.com/Davis-Replacement-Anemometer-Part-6410/dp/B000WBR1QS/ref=sr_1_1?ie=UTF8&qid=1398940677&sr=8-1&keywords=anemometer+davis).
- [26]. Aosong Electronics Co., Ltd. Datasheet DHT22. Datasheet DHT22. [En línea] [Citado el: 10 de 05 de 2014.] <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.

- [27]. STALLINGS, Willan. Comunicaciones y Redes de Computación. MADRID : PEARSON EDUCACIÓN, S.A., 2004.
- [28]. Sparkfu. [En línea] [Citado el: 06 de 06 de 2014.] <https://www.sparkfun.com/tutorials/253>.
- [29]. Proviento. ProViento S.A. Energías Renovables del Ecuador. [En línea] Olaf Schwetje, 2011. [Citado el: 02 de 09 de 2014.] www.proviento.com.ec/.
- [30]. Davis. DAVIS. DAVIS Instruments. [En línea] [Citado el: 08 de 06 de 2014.] <http://davisnet.com/>.
- [31]. VAISALA. [En línea] [Citado el: 18 de 9 de 2014.] <http://es.vaisala.com/sp/products/automaticweatherstations/Pages/MAWS201.aspx>.
- [32]. RDSI. Red Digital Servicios Integrados. [En línea] [Citado el: 08 de 10 de 2014.] <http://thales.cica.es/rd/Recursos/rd97/Otros/52-3-o-rdsi.html>.
- [33]. SISTEMAS TECNOLÓGICOS. [En línea] [Citado el: 07 de 10 de 2014.] <http://www.sistemastecnologicos.com.ec>.
- [34]. SÁNCHEZ, Pedro. Electrónica & Robótica. Configuración de resistencias en Pull-Up y Pull-Down. [En línea] 23 de 02 de 2013. [Citado el: 25 de 07 de 2014.] <http://electronicayrobotica.wordpress.com/2013/02/23/configuracion-de-resistencias-en-pull-up-y-pull-down/>.
- [35]. HUIDOBRO, José y CONESA, Rafael. Sistemas de telefonía . Madrid : Thomson Editors Spain Paraninfo, S.A., 2006. Quinta edición.
- [36]. SALLEN, Oriol; RAMON, Agusti; VALENZUELA, José. Principio de las comunicaciones móviles . Cataluña : Edicions UPC, 2003. Primera edición.

k ANEXOS

k.1 ANEXO N° 1 CIRCUITO DEL PROTOTIPO IMPLEMENTADO

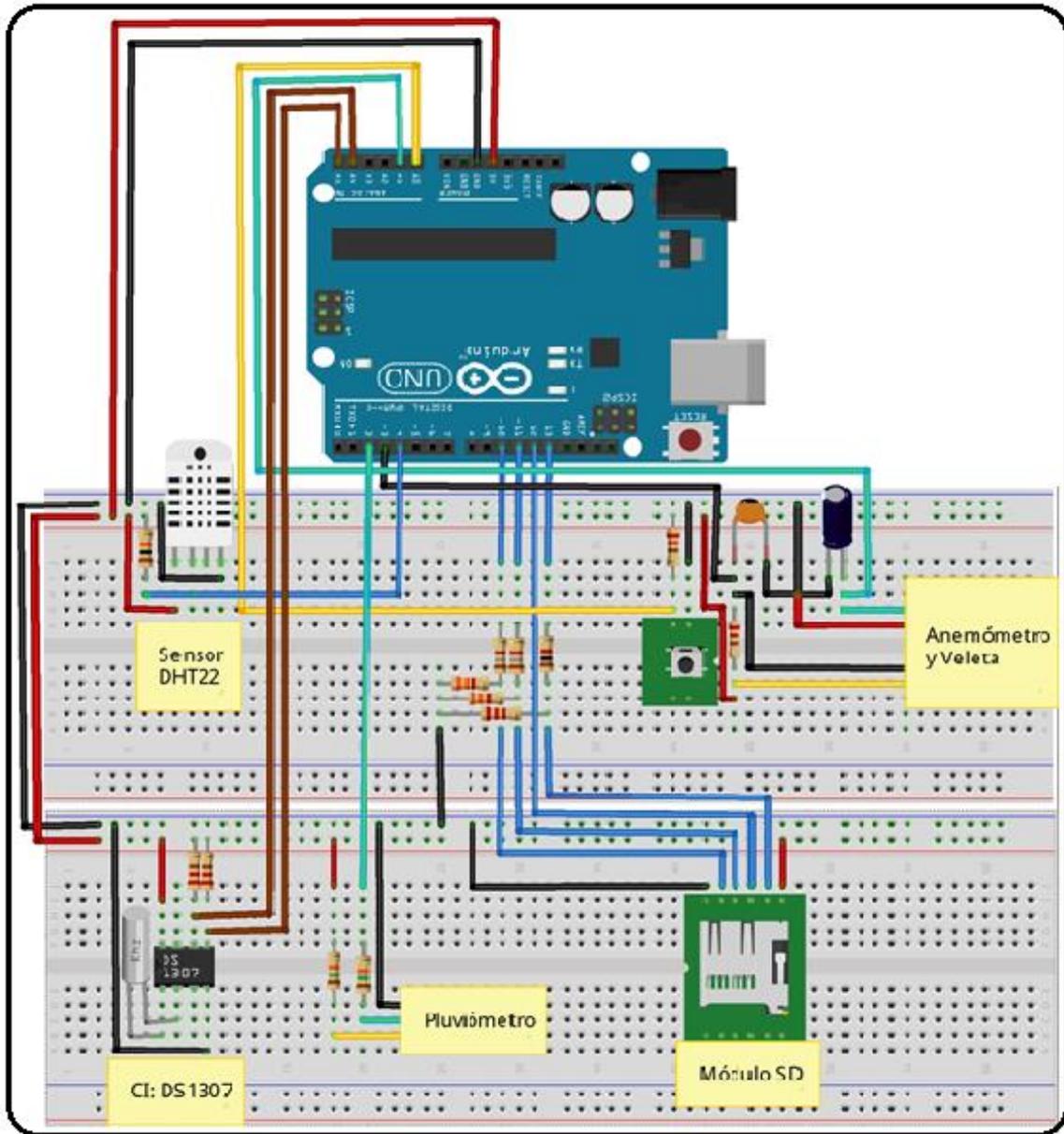


Figura 60: Esquema de conexión utilizado para la adquisición de las variables climatológicas, simulado en Fritzing. Fuente: (El Autor).

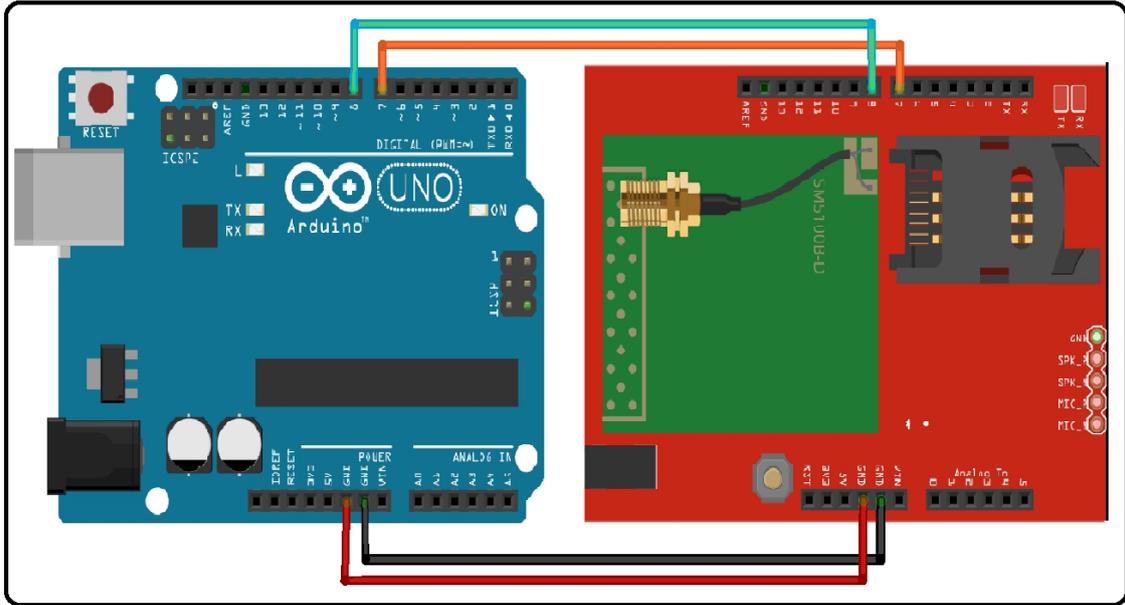


Figura 61: Esquema de conexión utilizado para la transmisión de los datos, del prototipo de estación meteorológica, simulado en Fritzing. Fuente: (El Autor).

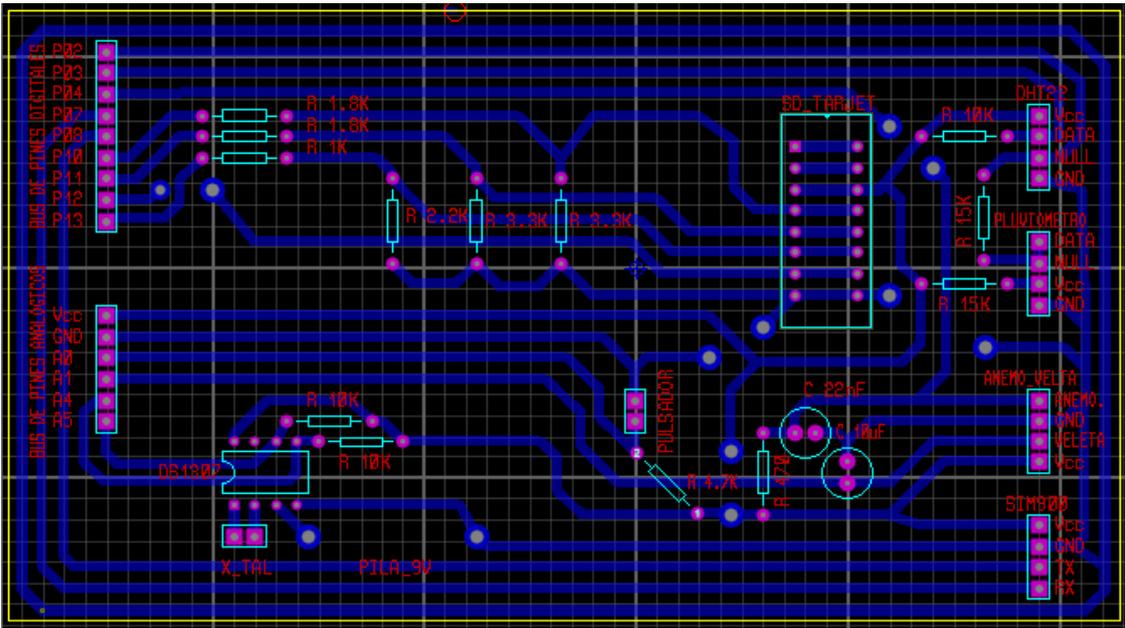


Figura 62: Diseño del circuito impreso para el acople de los distintos sensores utilizados, realizado en ARES. Fuente: (El Autor).

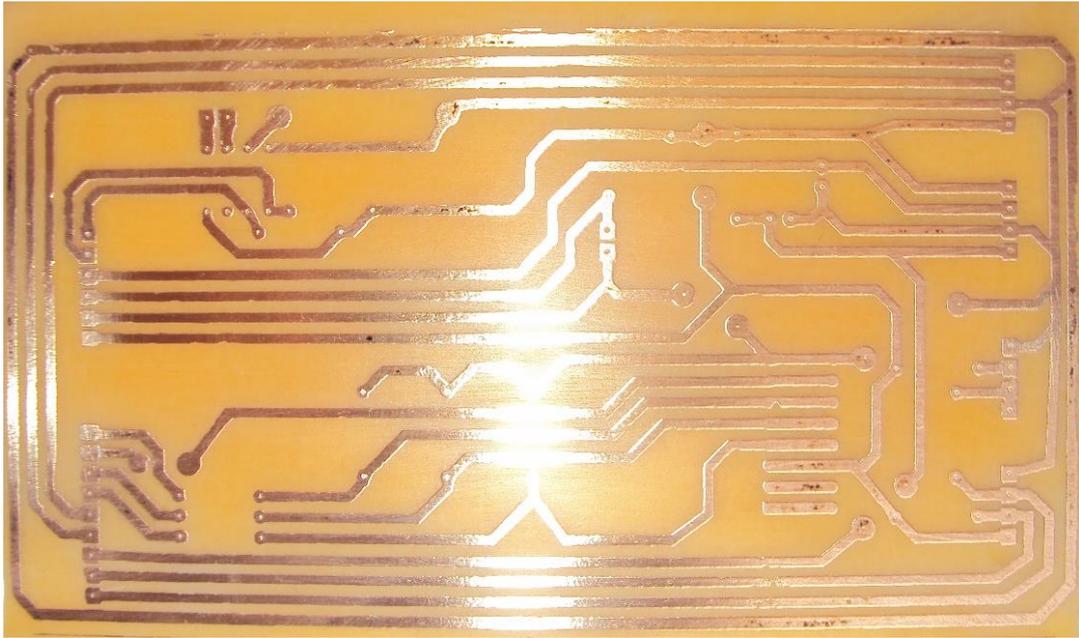


Figura 63: Gráfica de la placa impresa para la adecuación de RTC, SD Y SENSORES.
Fuente: (El Autor).



Figura 64: Gráfica de los sensores: DHT22, anemómetro, veleta y pluviómetro.
Fuente: (El Autor).



Figura 65: Gráfica del datalogger y la adecuación para los sensores. Fuente: (El Autor).

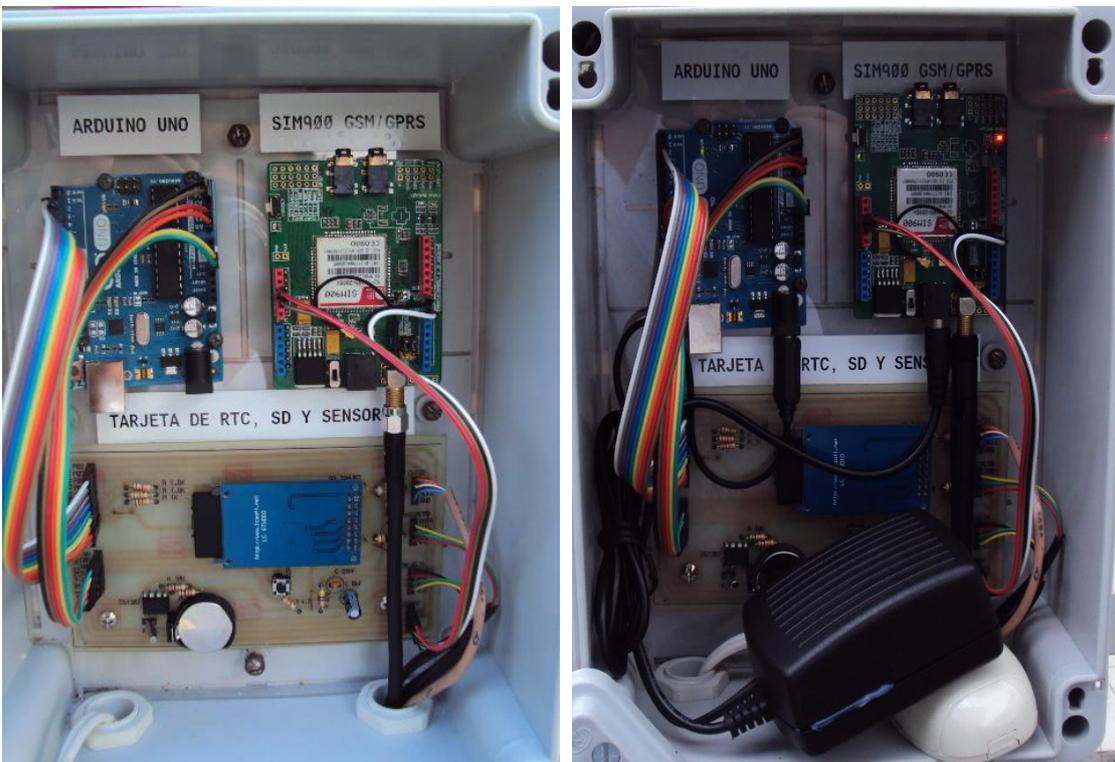


Figura 66: Gráfica del prototipo implementado. Fuente: (El Autor).

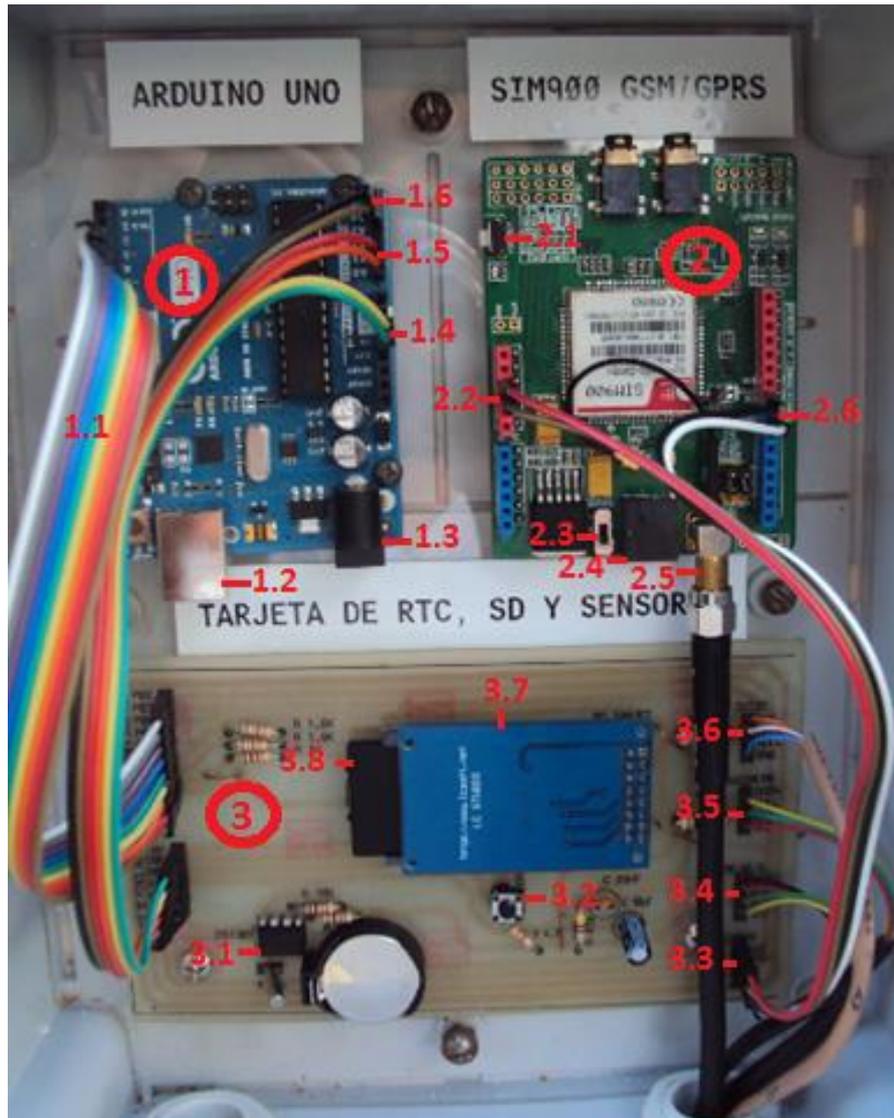


Figura 67: Indicaciones técnicas del prototipo de estación meteorológica.

Fuente: (El Autor).

1. Arduino Uno.

- 1.1. Bus de datos para almacenamiento y transmisión de variables Climáticas.
- 1.2. Puerto de carga.
- 1.3. Puerto de alimentación DC.
- 1.4. Alimentación a la tarjeta de RTC, SD y Sensores.
- 1.5. Calibración y lectura de Veleta.
- 1.6. Comunicación RTC (SDA, SCL).

2. Módulo SIM900 GSM/GPRS.

- 2.1. Pulsador de encendido y apagado del Módulo.
- 2.2. Alimentación desde Arduino.

- 2.3. Selector de alimentación.
- 2.4. Puerto de alimentación DC.
- 2.5. Conector de antena.
- 2.6. Comunicación con Arduino Uno (Tx, Rx).

3. Tarjeta de RTC, SD y Sensores.

- 3.1. Circuito de RTC.
- 3.2. Calibración de veleta.
- 3.3. Alimentación, Tx y Rx.
- 3.4. Sensor Veleta.
- 3.5. Sensor anemómetro.
- 3.6. Sensor de temperatura DHT22.
- 3.7. Módulo SD.
- 3.8. Slot de memoria microSD 2GB.

Tabla 17: Tabla de datos almacenados en la SD del prototipo de estación meteorológica. Fuente: (El Autor).

Temperatura	Humedad	Vel_Viento	Direc_Viento	Precipitación	Hora	Fecha
16.40	63.30	9.60	359	0.00	6:20	14/10/2014
16.40	63.30	10.30	340	0.00	6:20	14/10/2014
16.40	63.00	13.20	315	0.00	6:21	14/10/2014
16.40	63.10	4.80	339	0.00	6:21	14/10/2014
16.40	62.90	14.40	329	0.00	6:22	14/10/2014
16.40	63.70	12.00	340	0.00	6:22	14/10/2014
16.40	62.50	12.00	355	0.00	6:23	14/10/2014
16.50	63.30	7.20	326	0.00	6:23	14/10/2014
16.40	62.90	7.20	359	0.00	6:24	14/10/2014
16.40	63.30	8.40	334	0.00	6:24	14/10/2014
16.40	62.50	8.40	358	0.00	6:25	14/10/2014
16.50	63.40	8.40	321	0.00	6:25	14/10/2014
16.40	62.80	13.20	265	0.00	6:26	14/10/2014
16.40	63.30	14.00	280	0.00	6:26	14/10/2014
16.50	63.60	14.40	343	0.00	6:27	14/10/2014
16.50	63.40	14.40	356	0.00	6:27	14/10/2014
16.50	62.80	13.20	5	0.00	6:28	14/10/2014
16.50	62.90	7.20	359	0.00	6:28	14/10/2014
16.50	62.70	3.60	86	0.00	6:29	14/10/2014
16.60	62.80	3.60	54	0.00	6:29	14/10/2014
16.50	64.00	4.80	11	0.00	6:30	14/10/2014
16.50	64.30	3.60	359	0.00	6:30	14/10/2014
16.50	64.20	6.00	355	0.00	6:31	14/10/2014
16.40	63.30	4.60	355	0.00	6:31	14/10/2014
16.40	63.20	1.20	355	0.00	6:32	14/10/2014
16.40	62.40	9.60	191	0.00	6:32	14/10/2014
16.40	62.90	6.00	130	0.00	6:33	14/10/2014
16.50	62.90	7.20	82	0.00	6:33	14/10/2014
16.50	62.80	1.20	79	0.00	6:34	14/10/2014
16.50	63.40	0.00	78	0.00	6:34	14/10/2014
16.50	64.00	4.80	359	0.00	6:35	14/10/2014
16.50	63.30	6.60	359	0.00	6:35	14/10/2014
16.50	63.50	7.20	282	0.00	6:36	14/10/2014
16.50	64.20	18.00	359	0.00	6:36	14/10/2014
16.40	63.70	7.20	322	0.00	6:37	14/10/2014
16.50	64.40	9.60	326	0.00	6:37	14/10/2014
16.40	64.30	4.80	37	0.00	6:38	14/10/2014
16.40	64.50	7.20	358	0.00	6:38	14/10/2014
16.40	64.20	4.80	352	0.00	6:39	14/10/2014

16.40	63.30	7.60	359	0.00	6:39	14/10/2014
16.40	64.10	10.80	342	0.00	6:40	14/10/2014
16.40	64.00	7.20	67	0.00	6:40	14/10/2014
16.40	64.20	16.80	349	0.00	6:41	14/10/2014
16.40	64.60	7.20	7	0.00	6:41	14/10/2014
16.40	63.70	3.60	304	0.00	6:42	14/10/2014
16.40	64.50	6.00	298	0.00	6:42	14/10/2014
16.40	63.80	4.80	312	0.00	6:43	14/10/2014
16.40	64.30	1.20	312	0.00	6:43	14/10/2014
16.40	64.20	6.00	306	0.00	6:44	14/10/2014
16.40	64.20	6.00	310	0.00	6:44	14/10/2014
16.40	64.70	6.00	322	0.00	6:45	14/10/2014
16.40	65.40	7.20	280	0.00	6:45	14/10/2014
16.40	64.80	7.20	242	0.00	6:46	14/10/2014
16.30	64.70	8.40	204	0.00	6:46	14/10/2014
16.30	64.00	3.60	220	0.00	6:47	14/10/2014
16.40	64.40	7.20	359	0.00	6:47	14/10/2014
16.40	65.20	6.00	2	0.00	6:48	14/10/2014
16.40	63.30	6.00	280	0.00	6:48	14/10/2014
16.40	65.30	6.00	328	0.00	6:49	14/10/2014
16.40	64.30	6.00	324	0.00	6:49	14/10/2014
16.30	64.30	7.20	293	0.00	6:50	14/10/2014
16.30	65.00	3.60	325	0.00	6:50	14/10/2014
16.30	65.20	15.60	332	0.00	6:51	14/10/2014
16.30	65.80	12.00	289	0.00	6:51	14/10/2014
16.30	65.80	3.60	7	0.00	6:52	14/10/2014
16.40	63.30	9.60	359	0.00	6:20	14/10/2014
16.30	64.50	4.80	126	0.00	6:53	14/10/2014
16.30	64.70	4.80	340	0.00	6:53	14/10/2014
16.30	65.70	4.80	48	0.00	6:54	14/10/2014
16.30	66.10	2.40	47	0.00	6:54	14/10/2014
16.30	66.30	3.60	2	0.00	6:55	14/10/2014
16.30	65.70	3.60	335	0.00	6:55	14/10/2014
16.30	64.70	8.40	290	0.00	6:56	14/10/2014
16.30	63.90	2.40	292	0.00	6:56	14/10/2014
16.40	63.60	1.20	292	0.00	6:57	14/10/2014
16.40	63.60	1.60	292	0.00	6:57	14/10/2014
16.40	64.10	2.40	296	0.00	6:58	14/10/2014
16.40	65.00	7.20	312	0.00	6:58	14/10/2014
16.50	65.70	7.20	339	0.00	6:59	14/10/2014
16.50	61.80	3.60	316	0.00	6:59	14/10/2014
16.60	61.50	3.60	316	0.00	7:00	14/10/2014

k.2 ANEXO N° 2 ALGORITMOS DE PROGRAMACIÓN EN ARDUINO

ALGORITMO PARA LA CREACIÓN DE ARCHIVO DE LAS VARIABLES METEOROLÓGICAS.

```
File fp;                                // Crea un apuntador fp.
fp = SD.open("data.txt",FILE_WRITE);    // Crea un archivo de texto plano con el nombre de DATA.
if(fp) {
fp.print("Temperatura");                // Escribe el nombre de la variable a almacenar.
fp.print(" ");
fp.println();
fp.close();
}
```

ALGORITMO PARA LA LECTURA DE LA TEMPERATURA Y HUMEDAD.

```
float t, h;                              // Variables para sensor temperatura y humedad.
h = dht.readHumidity();                   // Función que devuelve un valor de la humedad en punto flotante.
t = dht.readTemperature();               // Función que devuelve un valor de la temperatura en punto flotante.
```

ALGORITMO PARA EL CÁLCULO DE LA VELOCIDAD DEL VIENTO.

```
loop(){
RPM_Tops = 0;
if ((RPM_Tops >= 0) and (RPM_Tops <= 21)) RPM = RPM_Tops * 1.2; // Rango de velocidad en milisegundos multiplicado por la constante para
if ((RPM_Tops > 21) and (RPM_Tops <= 45)) RPM = RPM_Tops * 1.15; // conversión a Km/h.
if ((RPM_Tops > 45) and (RPM_Tops <= 90)) RPM = RPM_Tops * 1.1;
if ((RPM_Tops > 90) and (RPM_Tops <= 156)) RPM = RPM_Tops * 1.0;
}
```

ALGORITMO PARA EL CÁLCULO DE LA DIRECCIÓN DEL VIENTO.

```
loop (){
PotValue = analogRead(PotPin); // Conversión ADC del sensor, para determinar la dirección del viento.
Direction = map(PotValue, 0, 1023, 0, 359); // Se establece el ADC en un rango de salida de 0 a 359.
convert: // Proceso para calcular la dirección del viento en grados tomando como referencia la dirección Norte.
if (Direction < 0) {Direction = Direction + 360; goto convert;} // Si se tiene un valor menor a 0 la dirección es 360 menos dicho valor.
```

```

if (Direction > 360) {Direction = Direction - 360; goto convert;} // Si se tiene un valor mayor a 0 la dirección el valor obtenido menos 360.
if ((Direction)<23) { Serial.println(" N");} // Rango que corresponde al "NORTE".
if ((Direction>22) && (Direction<68)) {Serial.println("NE");} // Rango que corresponde al "NORESTE".
if ((Direction>67) && (Direction<113)) {Serial.println(" E");} // Rango que corresponde al "ESTE".
if ((Direction>112) && (Direction<158)) {Serial.println("SE");} // Rango que corresponde al "SURESTE".
if ((Direction>157) && (Direction<203)) {Serial.println(" S");} // Rango que corresponde al "SUR".
if ((Direction>202) && (Direction<247)) {Serial.println("SW");} // Rango que corresponde al "SUROESTE".
if ((Direction>246) && (Direction<292)) {Serial.println(" W");} // Rango que corresponde al "OESTE".
if ((Direction>291) && (Direction<337)) {Serial.println("NW");} // Rango que corresponde al "NOROESTE".
if ((Direction>336) && (Direction<=360)){Serial.println(" N");} // Rango que corresponde al "NORTE".
}

```

ALGORITMO PARA EL CÁLCULO DE LA PRECIPITACIÓN.

```

contPluv () { // Función para contar y almacenar la cantidad de lluvia.
cambio_pluv++; // Incrementa y almacena cada vuelco producido en el pluviómetro.
}

float calcIntensidadLluvia() { // Función para calcular la intensidad de lluvia.

```

```
float x; // Variable para el cálculo de la cantidad de agua caída.  
x = cambio_pluv*0.20; // Cálculo de la cantidad de lluvia con el factor de conversión.  
}
```

ALGORITMO PARA LA TRANSMISIÓN DE LOS DATOS.

```
void SendData(){  
  GPRS.print("AT+HTTTPARA=\"URL\",\"emoncms.org/api/post?apikey=9bd92770fbff3c83dfdca522f9877de3&json={ }\"); // Establecer conexión http  
  / //con el servidor emoncms.  
  GPRS.print("Temperatura:"); // Se envía el valor de la variable sensada.  
  GPRS.print(t);  
  GPRS.print("}\r\n"); // se cierra la trama de datos.  
  delay(100);  
  Respond_GPRS(); // Se espera una respuesta del módulo SIM900.  
  GPRS.println("AT+HTTPACTION=1");  
}
```

FIN

k.3 ANEXO N° 3 HOJA DE DATOS DE ARDUINO UNO R3

Technical Specification

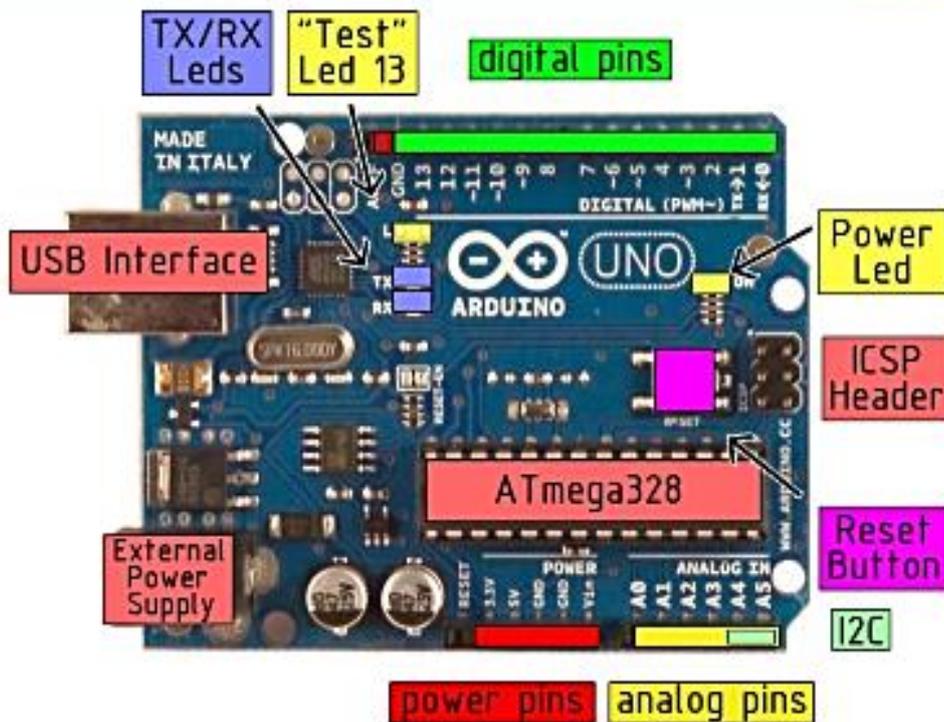


EAGLE files: [arduino-duemilerove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



radiospares RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Sketch - Arduino IDE
File Edit Sketch Tools Help
[Icons]
Sketch
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

Done compiling.
Press Compile button (to check for errors)

Upload

TX RX Flashing

Blinking Led!



radiospares

RADIONICS



k.4 ANEXO N° 4 HOJA DE DATOS DE MÓDULO SIM900 GSM/GRPS

Arduino GPRS Shield

Introduction



The GPRS Shield is based on SIM900 module from SIMCOM and compatible with Arduino and its clones. The GPRS Shield provides you a way to communicate using the GSM cell phone network. The shield allows you to achieve SMS, MMS, GPRS and Audio via UART by sending AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands). The shield also has the 12 GPIOs, 2 PWMs and an ADC of the SIM900 module (They are all 2V8 logic) present onboard.

Features

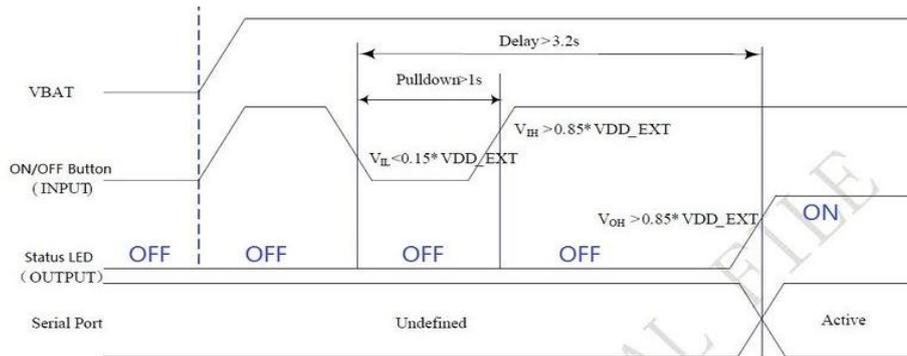
- Quad-Band 850 / 900/ 1800 / 1900 MHz - would work on GSM networks in all countries across the world.
- GPRS multi-slot class 10/8.
- GPRS mobile station class B.
- Class 4 (2 W @ 850 / 900 MHz).
- Class 1 (1 W @ 1800 / 1900MHz).
- Control via AT commands - Standard Commands: GSM 07.07 & 07.05 | Enhanced Commands: SIMCOM AT Commands.
- Short Message Service - so that you can send small amounts of data over the network.
- Embedded TCP/UDP stack - allows you to upload data to a web server.
- RTC supported.
- Selectable serial port.
- Low power consumption - 1.5mA (sleep mode).
- Industrial Temperature Range -40°C to +85 °C.

Power Up and Power Down the GPRS Shield

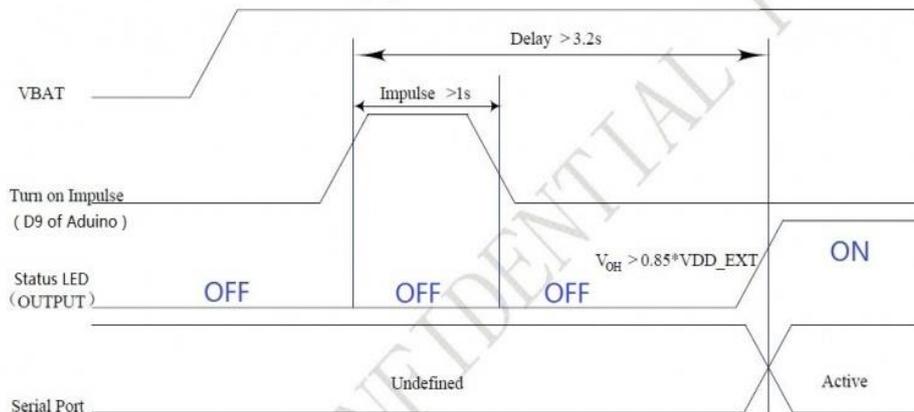
Power Up the GPRS Shield

The GPRS Shield can be turned on by two ways:

- 1, **Hardware Trigger**; Press the ON/OFF Button about two seconds. The power up scenarios illustrates as following figure:



- 2, **Software Trigger**; If use this way to power up the GPRS Shield, JP need to be soldered, then Digital Pin 9 of the Arduino will act as Software Trigger port and Digital Pin 9 can not be use as other purpose. Then give Digital Pin 9 a Turn on Impulse can power up the GPRS Shield. The power up scenarios illustrates as following figure:

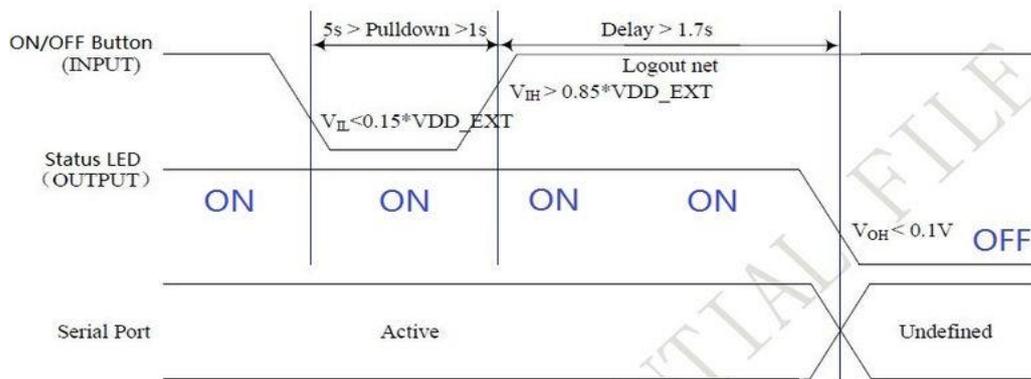


Power Down the GPRS Shield

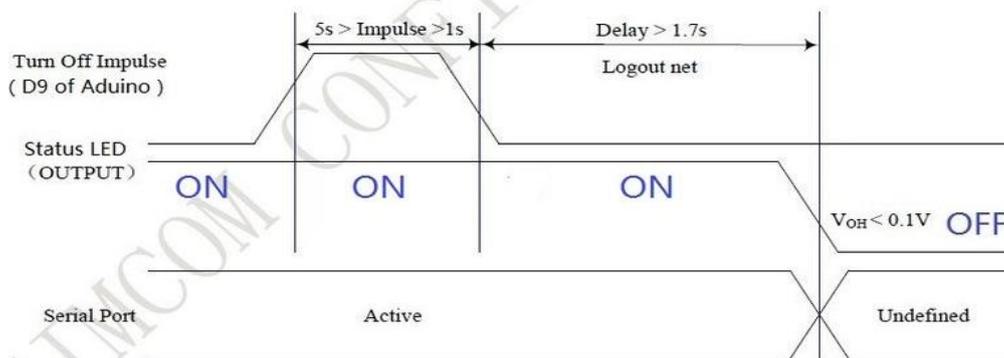
The GPRS Shield can be turned off by following ways:

- 1, **Normal power down procedure**: Turn off the GPRS shield by using Hardware Trigger; Press the ON/OFF Button about two seconds.

The power down scenarios illustrates as following figure:

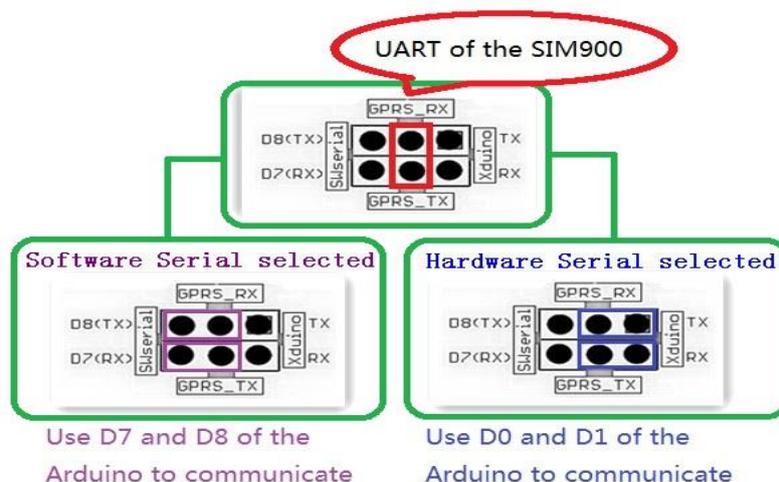


2, **Normal power down procedure:** If JP is soldered, then give Digital Pin 9 of the Arduino (act as Software Trigger) a Turn off Impulse can turn off the GPRS Shield. The power down scenarios illustrates as following figure:



Serial Port (UART) Communication.

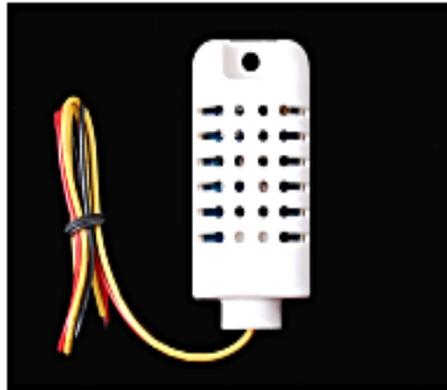
The GPRS Shield is used UART protocol to communicate with an Arduino/Arduino clone; Users can use jumpers to connect (RX, TX) of the shield to either Software Serial (D8, D7) or Hardware Serial (D1, D0) of the Arduino. Detailed information is showed as the following picture:



k.5 ANEXO N° 5 HOJA DE DATOS DEL SENSOR DHT22



Standard AM2302/DHT22



AM2302/DHT22 with big case and wires

Digital relative humidity & temperature sensor AM2302/DHT22

1. Feature & Application:

- *High precision
- *Capacitive type
- *Full range temperature compensated
- *Relative humidity and temperature measurement
- *Calibrated digital signal
- *Outstanding long-term stability
- *Extra components not needed
- *Long transmission distance, up to 100 meters
- *Low power consumption
- *4 pins packaged and fully interchangeable

2. Description:

AM2302 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

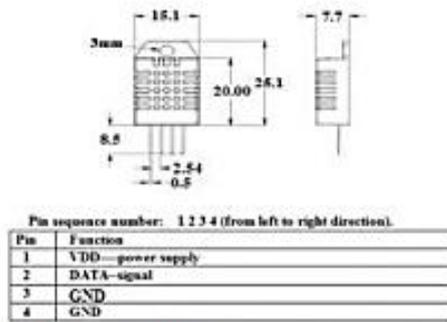
Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable AM2302 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

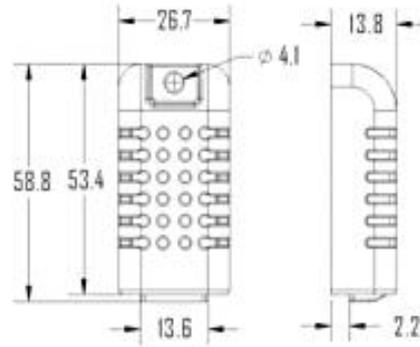
3. Technical Specification:

Model	AM2302
Power supply	3.3-5.5V DC
Output signal	digital signal via 1-wire bus
Sensing element	Polymer humidity capacitor
Operating range	humidity 0-100%RH; temperature -40-80Celsius
Accuracy	humidity +2%RH (Max +5%RH); temperature +0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +1%RH; temperature +0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Interchangeability	fully interchangeable

4. Dimensions: (unit—mm)



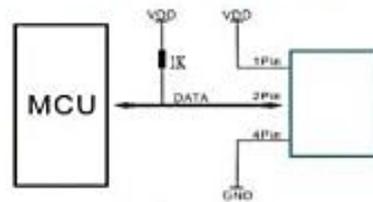
Standard AM2302's dimensions as above



Big case's dimensions as above

Red wire—power supply, Black wire—GND
Yellow wire—Data output

5. Electrical connection diagram:



6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

1-wire bus is used for communication between MCU and AM2302. (Our 1-wire bus is specially designed, it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of our 1-wire bus:

DATA=16 bits RH data+16 bits Temperature data+8 bits check-sum

Example: MCU has received 40 bits data from AM2302 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Here we convert 16 bits RH data from binary system to decimal system,

0000 0010 1000 1100 → 652

Binary system Decimal system

RH=652/10=65.2%RH

Here we convert 16 bits T data from binary system to decimal system,

0000 0001 0101 1111 → 351

Binary system Decimal system

T=351/10=35.1℃

When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius.

Example: **1000 0000 0110 0101**, T= minus 10.1℃

16 bits T data

Sum=0000 0010+1000 1100+0000 0001+0101 1111=**1110 1110**

Check-sum=the last 8 bits of Sum=1110 1110

k.6 ANEXO N° 6 HOJA DE DATOS DEL ANEMÓMETRO Y VELETA DAVIS

Anemometer



6410

Vantage Pro2™ Accessories

Includes both wind speed and wind direction sensors. Rugged components stand up to hurricane-force winds, yet are sensitive to a light breeze. Includes sealed bearings for long life. The range and accuracy specifications have been verified in wind-tunnel tests. In areas where icing of the anemometer is a problem, drip rings deflect water from the joint between moving parts.

General

Operating Temperature	-40° to +149°F (-40° to +65°C)
Sensor Type	
Wind Speed	Solid state magnetic sensor
Wind Direction	Wind vane and potentiometer
Attached Cable Length	40' (12 m)
Cable Type	4-conductor, 26 AWG
Connector	Modular connector (RJ-11)
Maximum Cable Length	240' (73 m)

Note: Maximum displayable wind speed decreases as cable increases. At 140' (42 m) of cable, maximum displayable wind speed is 135 mph (60 m/s); at 240' (73 m), maximum wind speed displayed is 100 mph (45 m/s).

Material	
Wind Vane and Control Head	UV-resistant ABS
Wind Cups	Polycarbonate
Anemometer Arm	Black-anodized aluminum
Dimensions (length x width x height)	15.0" x 1.5" x 16.0" (381 mm x 38 mm x 457 mm)
Weight	1 lbs. 4 oz. (1.332 kg)

Sensor Output

Wind Direction

Display Resolution	16 points (22.5°) on compass rose, 1° in numeric display
Accuracy	±3°

Wind Speed

Resolution and Units	Measured in 1 mph. Other units are converted from mph and rounded to nearest 1 km/h, 0.1 m/s, or 1 knot
Range	1 to 200 mph, 1 to 173 knots, 0.5 to 89 m/s, 1 to 322 km/h
Accuracy	±2 mph (2 kts, 3 km/h, 1 m/s) or ±5%, whichever is greater
Maximum Cable Length	240' (73 m). Maximum wind speed reading decreases as length of cable from Anemometer to ISS increases. At 140' (42 m), maximum speed is 135 mph (60 m/s). At 240', the maximum is 100 mph.

Input/Output Connections

Black	Wind speed contact closure to ground
Red	Ground
Green	Wind direction pot wiper (20KΩ potentiometer)
Yellow	Pot supply voltage
Wind Speed Translation Formula	1600 rev/hr = 1 mph $V = P(2.25/T)$ (V = speed in mph, P = no. of pulses per sample period T = sample period in seconds)
Wind Direction Translation	Variable resistance 0 - 20KΩ; 10KΩ = south, 160°

Package Dimensions

Product #	Package Dimensions (Length x Width x Height)	Package Weight	UPC Codes
6410	17.75" x 10.50" x 3.00" (451 mm x 267 mm x 76 mm)	2.0 lbs. (.9 kg)	011698 00237 5

DAVIS  **Davis Instruments** 3465 Diablo Ave., Hayward, CA 94545-2776 USA
 (510) 732-9229 • FAX (510) 670-0589 • sales@davisnet.com • www.davisnet.com

DS6410 (Rev. H, 2/1/13)

Specifications

General

Sensor Type	Tipping bucket with magnetic reed switch
Output	Contact closure
Attached Cable Length	40' (12 m)
Cable Type	4-conductor, 26 AWG
Connector	Modular connector (RJ-11)
Recommended Maximum Cable Length	900' (270 m)
Housing Material	UV-stabilized ABS plastic
Dimensions	
Rain Collector	8.75" diameter x 9.5" high (16.5 cm diameter x 24 cm high)
Collection Area	33.2 in ² (214 cm ²)
Weight	2 lbs. 3 oz. (1 kg)

Console Data

These specifications apply to sensor output as converted by Davis Instruments weather station consoles.

Range

Daily Rainfall	0.00" to 99.99" (0.0 mm to 999.8 mm)
Total Rainfall	0.00" to 99.99" (0.0 mm to 9999 mm)

Accuracy

Rainfall	±3%, ±1 rainfall count between 0.01" and 2.00" per hour (0.2 mm and 50.0 mm per hour); ±5%, ±1 rainfall count between 2.00" and 4.00" per hour (50.0 mm and 100.0 mm per hour)
Resolution	0.01" (0.2 mm)
Sample and Display Update Interval	16 seconds (max)

WeatherLink® Data

These specifications apply to sensor output as logged and displayed by the WeatherLink.

Daily Rainfall	Total during archive interval
Total Rainfall	Total during archive interval
Rate of Rainfall	Maximum value during archive interval (For Vantage Pro and Pro2 models only)

Input/Output Connections

Red	Switch terminal
Green & Yellow	Switch terminal

k.8 ANEXO N° 8 HOJA DE DATOS DEL CIRCUITO INTEGRADO DS1307 O RTC



DS1307 64 x 8 Serial Real-Time Clock

www.maxim-ic.com

FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

DESCRIPTION

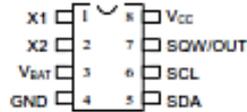
The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

ABSOLUTE MAXIMUM RATINGS*

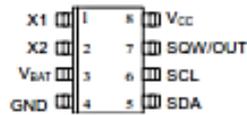
Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds DIP See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)



DS1307 8-Pin SOIC (150-mil)

PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

Range	Temperature	V _{CC}
Commercial	0°C to +70°C	4.5V to 5.5V V _{CC1}
Industrial	-40°C to +85°C	4.5V to 5.5V V _{CC1}

RECOMMENDED DC OPERATING CONDITIONS

(Over the operating range*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	
Logic 1	V_{IH}	2.2		$V_{CC} + 0.3$	V	
Logic 0	V_{IL}	-0.5		+0.8	V	
V_{BAT} Battery Voltage	V_{BAT}	2.0		3.5	V	

*Unless otherwise specified.

DC ELECTRICAL CHARACTERISTICS

(Over the operating range*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage (SCL)	I_{LI}			1	μ A	
I/O Leakage (SDA & SOW/OUT)	I_{LO}			1	μ A	
Logic 0 Output ($I_{OL} = 5$ mA)	V_{OL}			0.4	V	
Active Supply Current	I_{CCA}			1.5	mA	7
Standby Current	I_{CCS}			200	μ A	1
Battery Current (OSC ON); SOW/OUT OFF	I_{BAT1}		300	500	nA	2
Battery Current (OSC ON); SOW/OUT ON (32kHz)	I_{BAT2}		480	800	nA	
Power-Fail Voltage	V_{PF}	$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$	V	8

*Unless otherwise specified.

AC ELECTRICAL CHARACTERISTICS

(Over the operating range*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	kHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			μ s	
Hold Time (Repeated) START Condition	t_{HD-STA}	4.0			μ s	3
LOW Period of SCL Clock	t_{LOW}	4.7			μ s	
HIGH Period of SCL Clock	t_{HIGH}	4.0			μ s	
Set-up Time for a Repeated START Condition	t_{SU-STA}	4.7			μ s	
Data Hold Time	t_{HD-DAT}	0			μ s	4.5
Data Set-up Time	t_{SU-DAT}	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set-up Time for STOP Condition	t_{SU-STO}	4.7			μ s	
Capacitive Load for each Bus Line	C_B			400	pF	6
I/O Capacitance ($T_A = 25^\circ\text{C}$)	C_{IO}		10		pF	
Crystal Specified Load Capacitance ($T_A = 25^\circ\text{C}$)			12.5		pF	

*Unless otherwise specified.



Washington
ENGLISH INSTITUTE

Una metodología, personalizada y eficaz de aprender Inglés



Washington
ENGLISH INSTITUTE

CERTIFICACION

WASHINGTON ENGLISH INSTITUTE.

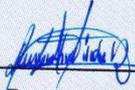
CERTIFIES:

Que el presente documento titulado "DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA, CON UN SISTEMA DE TELEMETRÍA PARA EL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA." está correctamente traducido del idioma español al idioma inglés, sin haber cambiado, aumentado o disminuido su sentido en ninguna línea o párrafo del mismo.

LO CERTIFICO:

Loja, 20 de Febrero de 2015




Lic. Dunia Vivanco V.
ESL teacher