

Universidad del Valle de Guatemala
Facultad de Ingeniería



**Implementación de Módulo de Predicción de
Tormentas en la Torre Meteorológica**

Trabajo de Graduación presentado por Jonathan Alejandro Mansilla
Gaitán para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala

2012

Implementación de Módulo de Predicción de Tormentas en la Torre Meteorológica

Universidad del Valle de Guatemala
Facultad de Ingeniería



**Implementación de Módulo de Predicción de
Tormentas en la Torre Meteorológica**

Trabajo de graduación presentado por Jonathan Alejandro Mansilla
Gaitán para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala

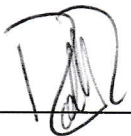
2012

Vo.Bo.

(f)  _____

Ing. Roberto Delgado

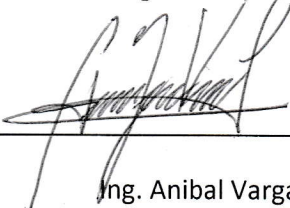
Tribunal Examinador:

(f)  _____

Ing. Roberto Delgado

(f)  _____

Ing. Luis Furlán

(f)  _____

Ing. Anibal Vargas

Fecha de aprobación : Guatemala, 22 de junio de 2012

ÍNDICE

ÍNDICE DE TABLAS	vii
ÍNDICE DE GRÁFICOS	viii
I. INTRODUCCIÓN	1
A. Objetivos del módulo	3
1. General del módulo.....	3
2. Específicos del módulo	3
II. DELIMITACIÓN.....	5
III. METODOLOGÍA.....	7
IV. MARCO TEÓRICO.....	9
A. Campo electroestático	9
B. Protocolo de comunicación RS-232.....	10
1. Especificaciones generales.....	11
2. Especificaciones eléctricas.	12
3. Especificaciones funcionales.	12
C. Protocolo de comunicación RS-485.....	15
1. Especificaciones funcionales.	16
D. Página web.....	18
1. HTML	19
2. PHP.....	19
E. Base de datos.....	20
V. COMUNICACIÓN SENSOR TORRE	21
A. Módulo principal o maestro	21
B. Sensor PreviStorm	24
C. Establecer comunicación entre el sensor y el módulo maestro.....	28

1. En la práctica	28
VI. COMUNICACIÓN TORRE SERVIDOR	41
A. Software de comunicación.....	41
B. Base de datos.....	43
C. Página web.....	45
D. Historial.....	46
E. Complementos	47
VII. CONCLUSIONES.....	49
VIII. RECOMENDACIONES.....	51
IX. BIBLIOGRAFÍA.....	53
X. ANEXOS	55
A. Anexo 1 – Código Fuente Microcontroladores	55
1. Código Fuente Módulo Principal	55
2. Código Fuente Módulo Detección de Tormentas	60
B. Anexo 2 – Código Fuente Módulo Comunicación y Página Web	61
1. Código Página Estado Actual.....	61
2. Código Página Historial.....	78
XI. GLOSARIO	89
XII. FICHA TÉCNICA DEL SISTEMA	93

ÍNDICE DE TABLAS

Tabla 1. Función de pines DB-9.....	13
Tabla 2. Señales Bus RS-485 de 2 hilos.....	17
Tabla 3. Señales Bus RS-485 de 4 hilos.....	18
Tabla 4. Pines utilizados.....	22
Tabla 5. Estado de la tormenta.....	28
Tabla 6. Estados driver MAX485.....	30
Tabla 7. Características del servidor.....	41
Tabla 8. Cadena – Trama de envío.....	42
Tabla 9. Estructura de la tabla mediciones.....	44
Tabla 10. Especificaciones técnicas ⁽¹⁵⁾	93
Tabla 11. Tipos de alarmas ⁽⁹⁾	97

ÍNDICE DE GRÁFICOS

Figura 1. Conector DB-9	12
Figura 2. Señales conector DB-9 (2)	13
Figura 3. Bus RS-485 de 2 hilos (4)	17
Figura 4. Bus RS-485 de 4 hilos (4)	18
Figura 5. Funciones módulo maestro	22
Figura 6. Diagrama circuito módulo maestro	23
Figura 7. UPS PreviStorm	25
Figura 8. MAD PreviStorm.....	25
Figura 9. MCE PreviStorm.....	27
Figura 10. Niveles eléctricos RS-232 Vs. Dispositivos TTL o CMOS	29
Figura 11. Comunicación maestro - MAD	31
Figura 12. Integración maestro - Esclavo.....	32
Figura 13. Circuito módulo de detección de tormentas (Versión inicial).....	33
Figura 14. Placa circuito	34
Figura 15. Inversor de voltaje (Transistor NPN)	35
Figura 16. Voltaje de entrada Vs. voltaje de salida (Inversor)	36
Figura 17. Circuito módulo de detección de tormentas (Versión mejorada).....	37
Figura 18. Placa circuito (Versión mejorada).....	38
Figura 19. Funciones módulo detección de tormentas.....	40
Figura 20. Software de comunicación	43
Figura 21. Ventana emergente campo electrostático	46
Figura 22. (A) Sensor MCE ⁽⁹⁾	94
Figura 23. (C) Módulo de adquisición – MAD ⁽⁹⁾	95
Figura 24. Interconexión MCE-MAD ⁽⁹⁾	96
Figura 25. Puertos de alimentación, comunicación y alarma ⁽⁹⁾	96

RESUMEN

La torre meteorológica es un sistema de recolección de información importante que permite observar y analizar el estado del clima en la región donde está ubicada la Universidad del Valle de Guatemala. Dicho sistema actualmente puede realizar la medición de ciertas variables climatológicas pero todavía existen muchas más que pueden ser medidas y estudiadas. Tal es el caso del campo electrostático del ambiente, el cual puede brindar información acerca de la aproximación, formación o estado de una tormenta.

El módulo de detección de tormentas busca medir el campo electrostático del ambiente logrando agregarle al sistema una importante variable a estudiar, ampliando de esta forma la información climática del sistema. Esta implementación permite el constante monitoreo de dicha variable en cualquier momento y en cualquier lugar aun cuando no se esté en frente del módulo de adquisición de datos, esto gracias al servidor y página web de la torre meteorológica.

I. INTRODUCCIÓN

La importancia de conocer las características y comportamiento del clima hoy en día nos ayudan para la prevención de desastres así también como para realizar estudios climatológicos con el fin de conocer el comportamiento del clima en una zona en especial donde se pretenda llevar a cabo cualquier actividad social, económica o ambiental. Es por eso que las mediciones realizadas por la torre meteorológica son indispensables para la realización de estudios que ayuden a la predicción climatológica y a otras aplicaciones de interés.

De la necesidad de ampliar las variables climatológicas a medir por parte de la torre meteorológica, se propuso la implementación de un sensor capaz de medir el campo electrostático del ambiente. Para ello, es necesario comunicar al sensor especialmente diseñado para realizar esta medición, con el sistema actual de la torre utilizando para ello un microcontrolador y otros componentes electrónicos necesarios. Debido a la importancia de esta variable para la predicción de tormentas, es necesario que el comportamiento de la misma pueda ser observado y analizado de manera remota para evitar la necesidad de tener que estar siempre cerca del módulo de adquisición de datos.

El módulo de detección de tormentas ofrecerá entonces, una herramienta más al sistema de la torre meteorológica permitiendo la visualización y búsqueda de las mediciones del campo electrostático en todo lugar y momento de manera ordenada, facilitando su análisis y estudio.

II. Objetivos del módulo

A. General del módulo

- Diseñar e Implementar un módulo capaz de brindar información útil para la predicción de tormentas mediante la medición del campo electrostático del lugar donde se encuentre ubicado.

B. Específicos del módulo

- Aumentar el rango de variables a medir por la Torre Meteorológica.
- Realizar la comunicación entre el sensor y el servidor.
- Convertir los datos recibidos a la variable climática de interés.
- Transmitir los datos recopilados hacia la base de datos del servidor destinado.
- Tener acceso a los datos recopilados por el sensor desde cualquier lugar y en cualquier momento.

III. DELIMITACIÓN

La implementación del módulo de detección de tormentas está enfocada en poder proveer un sistema capaz de medir constantemente la actividad eléctrica del medio ambiente. Con el fin predecir la formación o existencia de una tormenta eléctrica, gracias a los datos recopilados, almacenados y actualizados en la web. Para esto es necesaria la integración de un nuevo sensor con la actual torre meteorológica.

La delimitación del proyecto es experimental, con el ideal de poder seguir desarrollando, año tras año, una torre meteorológica más compleja, robusta y útil. Este trabajo se puede considerar como un módulo adicional que se le agregará al Megaproyecto de Torre Meteorológica. Dicho módulo se desarrollará y se pondrá a prueba con el fin de dejarlo implementado. Brindándole a la torre meteorológica una variable climatológica más que podrá ser medida y estudiada.

Es importante mencionar que las mediciones que la torre meteorológica recopila, están limitadas a las condiciones climáticas y a las especificaciones de funcionamiento de sus sensores y componentes. En dado caso, por ejemplo, de ser necesario medir un valor de campo electrostático mayor o menor a los que la misma podrá registrar, será necesaria la implementación de otro sensor más robusto. Algo que está planeado realizarse a lo largo de la implementación del actual módulo de detección de tormentas.

Para el desarrollo e implementación de este proyecto se utilizará únicamente un microcontrolador (dsPIC 30F4011), con el hardware necesario para recibir la información del sensor provisto por el departamento de Ingeniería Mecatrónica y transmitirla al microcontrolador principal (PIC16F887) del sistema. Esta sección será un “puente”, que permitirá la comunicación del sensor PreviStorm con el módulo principal de la torre meteorológica.

Luego se deberá manejar la comunicación entre el dsPIC 30F4011 (esclavo) y el PIC16F887 (maestro) a nivel de software, utilizando la implementación física de los drivers y protocolos RS-232 y RS-485. Aquí se espera que el microcontrolador

maestro (PIC16F887) pueda recibir la información transmitida por el microcontrolador esclavo (dsPIC 30F4011) y unificarla a la base de datos con el propósito de transmitirla por completo al servidor que almacenara dichos datos.

Finalmente se modificará el programa escrito en lenguaje C# y la página web, para que el servidor también reciba y almacene los datos provistos por el sensor PreviStorm, como lo hace con los otros sensores. El software de comunicación escrito en lenguaje C# se encargará de recibir la información recaudada y transmitida por el microcontrolador maestro (PIC16F887) con el fin de interpretarla y convertirla al valor de la variable de interés y almacenarla. La página web de la torre meteorológica se modificará para que la misma despliegue la nueva variable de campo electrostático.

IV. METODOLOGÍA

Para cumplir los objetivos planteados se debe obtener la mayor información posible acerca de todo el sistema de la torre meteorológica, en especial del módulo principal, así como del módulo de comunicación y página web.

Los datos provistos por el sensor PreviStorm son almacenados temporalmente por medio de un microcontrolador, el cual se encarga de obtener la información acerca del campo electrostático y transmitirla en el momento adecuado al módulo de comunicación y página web de la torre meteorológica. Para ello no solo es necesaria la modificación del software del microcontrolador principal o maestro, sino también hardware para realizar la comunicación y conversión del protocolo con el que se comunica el sensor (RS-232) al protocolo que maneja el modulo maestro (RS-485) para comunicarse con los microcontroladores esclavos.

Realizada la comunicación del sensor, por medio de un microcontrolador esclavo, con el mando central del sistema, es necesario que el software del módulo de comunicación también extraiga este nuevo dato, realice la conversión de bytes y lo almacene, como lo hace con el resto de las mediciones de las variables climáticas, por lo que se le agregarán las modificaciones correspondientes al actual software encargado de esto.

Debido a que la torre meteorológica ahora maneja una variable climática más, se debe adecuar de igual forma la página web para que esta nueva variable climática pueda ser observada y estudiada utilizando esta herramienta tal como se hace con el resto de variables climáticas que mide el sistema.

V. MARCO TEÓRICO

A. Campo electroestático

Una *tormenta* es un fenómeno caracterizado por la coexistencia de dos o más masas de aire de diferentes temperaturas. Aunque científicamente se define como tormenta a aquella nube capaz de producir un trueno audible, también se denominan tormentas a los fenómenos atmosféricos que en la tierra están asociados a lluvia, hielo, granizo, electricidad, nieve o fuertes vientos.

El contraste térmico y otras propiedades de las masas de aire húmedo dan origen al desarrollo de fuertes movimientos ascendentes y descendentes (convección) produciendo una serie de efectos característicos, como fuertes lluvias, vientos en la superficie e intensas descargas eléctricas. El ciclo de actividad de una tormenta típica presenta una fase inicial de formación, intermedia de madurez y final de decaimiento. (4)

La actividad eléctrica de una tormenta se representa por medio del campo electrostático, el cual puede provocar la formación de rayos si supera un determinado valor. El campo electrostático se expresa en unidades de kV/m ya que mide la diferencia de potencial existente entre la nube y el potencial cero de la tierra por unidad de longitud.

Las cargas eléctricas no precisan de ningún medio material para influir entre ellas y por ello las fuerzas eléctricas son consideradas fuerzas de acción a distancia. La acción a distancia se explica mediante efectos provocados por la entidad causante de la interacción, sobre el espacio mismo que la rodea, permitiendo asignar a dicho espacio, como por ejemplo el aire, propiedades medibles. (7)

Desde la aproximación o la formación de una tormenta, el campo electrostático a nivel de tierra varía de forma importante. En una situación normal

(terreno plano y tiempo soleado) el campo electrostático tiene un valor medio de 0.120kV/m. En presencia de una tormenta, el campo puede crecer hasta 14kV/m aproximadamente. (10)

Se recurre al concepto de campo electrostático para facilitar la descripción, en términos físicos, de la influencia que una o más cargas ejercen sobre el espacio que las rodea. (4)

Hoy en día, cada carga altera su entorno, impregnándolo de cierta propiedad que convencionalmente se denomina campo eléctrico, el cual va llenando paulatinamente todo el espacio, a una velocidad igual a la de la luz, y que es este campo el que luego interactúa con cualquier otra carga. Cuando este campo ha interesado todo el espacio, pasa a denominarse campo electrostático. (16)

Dicho esto, para apreciar en todos los puntos del espacio un campo electrostático habría que esperar un tiempo infinito, sin embargo, si se toma en consideración una región limitada alrededor de la carga, el campo electrostático se podrá apreciar en un tiempo aceptable. (16)

B. Protocolo de comunicación RS-232

El puerto serial RS-232C, presente en todas las computadoras actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre computadoras. Se llama serial porque los bits se reciben uno detrás de otro o “en serie”. El RS-232C es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas). (3)

El protocolo RS-232 es una norma o estándar mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de

cable permitido, las distancias entre equipos, los conectores, etc. Además de las líneas de transmisión (Tx) y recepción (Rx), las comunicaciones seriales poseen otras líneas de control de flujo, donde su uso es opcional dependiendo del dispositivo a conectar. (6)

La interfaz puede trabajar en comunicación asíncrona o síncrona y tipos de canal simplex, *half duplex* o *full duplex*. En un canal simplex los datos siempre viajarán en una dirección. En un canal *half duplex*, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección. En un canal *full duplex*, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de *handshaking* de la RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado. (1)

Si un dispositivo de los que están conectados a una interfaz RS-232 procesa los datos a una velocidad menor de la que los recibe, deben de conectarse las líneas *handshaking* que permiten realizar un control de flujo tal que al dispositivo más lento le dé tiempo de procesar la información. Las líneas de "*handshaking*" que permiten hacer este control de flujo son las líneas RTS y CTS. Los diseñadores del estándar no concibieron estas líneas para que funcionen de este modo, pero dada su utilidad en cada interfaz posteriormente se incluye este modo de uso. (14)

1. **Especificaciones generales.** El estándar RS-232 se basa en comunicación asíncrona es decir que los datos pueden ser transmitidos en cualquier momento por lo que deben tomarse precauciones para sincronizar la transmisión y recepción. En la comunicación serial se distinguen dos tipos de dispositivos: Los equipos terminales de datos DTE ("Data Terminal Equipment"), y los equipos de comunicación de datos DCE ("Data Communication Equipment").

En la terminología de las comunicaciones seriales se denomina "Sexo" de un equipo a su tipo; en este sentido DTE y DCE son "sexos" contrarios. A efectos

prácticos el sexo del equipo determina que asignación de señales tendrá su conector externo. En un extremo del cable de comunicaciones se encuentra un conector llamado DB-9 hembra y del otro extremo un conector DB-9 macho. Se llama así porque tiene la forma de una D, y tiene el número 9 porque tiene 9 pines o entradas. El estándar define que el conector hembra se situará en los DCE y el macho en el DTE. (12)

Figura 1. Conector DB-9



La otra versión de conector que utiliza la interfaz RS-232 es el conector DB-25.

2. **Especificaciones eléctricas.** Las señales con las que trabaja este puerto serial son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es un 1 lógico. La interfaz RS-232 está diseñada para operar en distancias cortas, la norma dice que es posible tener cables de hasta 15 metros empleando una velocidad de comunicación no mayor a 20kilobits/segundo. A pesar de esto, muchas veces se utilizan velocidades mayores con resultados aceptables. (12)

3. **Especificaciones funcionales.** El RS-232C consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9 más difundida. En cualquier caso, las computadoras no suelen emplear más de 9 pines en el conector DB-25. Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. A continuación se muestra la función de cada pin: (12)

Figura 2. Señales conector DB-9 (3)

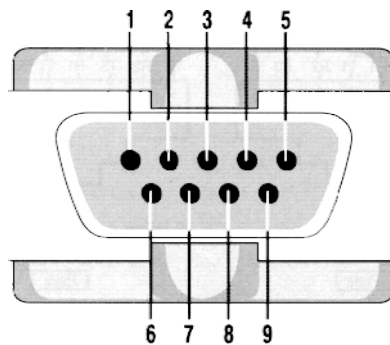


Tabla 1. Función de pines DB-9

No. Pin	Señal
1	Data Carrier Detect (DCD)
2	Received Data (RXD)
3	Transmitted Data (TXD)
4	Data Terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

DCD (Data Carrier Detect): Es la línea que indica que el modem tiene de verdad conexión remota. Este pin es también llamado "Receiver Time Signal Detect", es encendido cuando el modem recibe una señal remota y se mantiene encendido durante el enlace. En conexiones de modem *half duplex* el DCD es encendido solamente por el MODEM que está en recepción. (3)

RXD (Receive Data): Es la línea de recepción de datos serie desde el modem. (3)

TXD (Transmit Data): Es la línea de transmisión de datos serie al modem.
(3)

DTR (Data Terminal Ready): Esta señal prepara al modem para conectarse a una línea de comunicación y mantener la conexión establecida. Una vez que el modem está conectado a la línea este pin debe estar encendido para mantenerse la conexión. DTR también puede mantenerse encendido para efectuar la transmisión de datos por el pin TXD. (3)

SG (Signal ground): Esta línea debe estar conectada al chasis del PC y desde ahí hacer tierra. (3)

DSR (Data Set Ready): Es la línea que indica que el modem está preparado. Usualmente el DSR está todo el tiempo encendido porque este valor es un indicador de que el MODEM está encendido y listo. (3)

RTS (Request To Send): Es la línea que dice al modem que la PC quiere enviar datos. El estándar RS-232 dice que RTS condiciona al modem para la transmisión, en realidad esta es solo una función que sirve de interruptor a un modem *half duplex* para transmitir o recibir. Mientras un modem *half duplex* está recibiendo, el DTE guarda el RTS apagado (cero), Cuando le toca al DTE el turno de transmitir, este informa al modem que desea transmitir poniendo el RTS a (uno). El DTE no puede comenzar inmediatamente con el envío de datos al modem porque este, no puede cambiar inmediatamente el modo de transmisión a recepción. Después de poner a uno RTS el DTE monitorea el pin CTS el cual es apagado por el modem que está en modo de recepción, cuando el modem está listo para transmitir, este enciende el CTS en función al DTE que está listo para el envío de datos. Este RTS-CTS (*handshaking*), también es permitido en sentido contrario, es decir cuando el transmisor vuelve a recibir. Como en una comunicación *full duplex* hay dos canales no se necesita el RTS-CTS, de esta

forma un modem *full duplex* pone permanentemente el pin CTS conectado al pin DCD (Data Carrier Detect). (3)

CTS (Clear To Send): Es la línea que indica que el modem está preparado para recibir datos desde la PC. (3)

RI (Ring Indicator): Es la línea que indica que el modem ha detectado la señal de "llamada" (1 lógico). (3)

Además para que dos dispositivos puedan hacer efectivo el intercambio de información, se requiere que cada uno de ellos utilice las mismas características de transmisión, entre estas características están la velocidad de transmisión, que pueden ser de: 110bps, 300bps, 600bps, 900bps, 1200bps, 2400bps, 4800bps, 9600bps, 19200bps. (3)

C. Protocolo de comunicación RS-485

También conocido como TIA/EIA-485, es un estándar que define las características eléctricas de los conductores y receptores para su uso en sistemas digitales multipunto. La norma es publicada por la TIA/ EIA, Telecommunications Industry Association / Electronic Industries Alliance, por sus siglas en ingles. (15)

El estándar RS-485 está definido como un sistema en bus de transmisión multipunto diferencial, puede transmitir a altas velocidades sobre largas distancias (35 Mbps hasta 10 metros y 100 Kbps en 1.200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión. El medio físico de transmisión es un par entrelazado, con una longitud máxima de 1.200 metros operando entre 300 y 19200 bps y la comunicación *half duplex*. Soporta 32 transmisiones y 32 receptores. La transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto. Al tratarse de un estándar bastante abierto permite muchas utilizaciones y muy diferentes configuraciones.

La interfaz RS485 ha sido desarrollada, analógicamente a la interfaz RS-422, para la transmisión en serie de datos de alta velocidad a grandes distancias y encuentra creciente aplicación en el sector industrial. Pero mientras que la RS-422 sólo permite la conexión unidireccional de hasta 10 receptores en un transmisor, la RS-485 está concebida como sistema Bus bidireccional con hasta 32 participantes. (15)

Dado que varios transmisores trabajan en una línea común, tiene que garantizarse con un protocolo, que en todo momento esté activo como máximo un transmisor de datos. Los otros transmisores tienen que encontrarse en ese momento en estado de alta impedancia, o dicho en palabras sencillas, en reposo. (15)

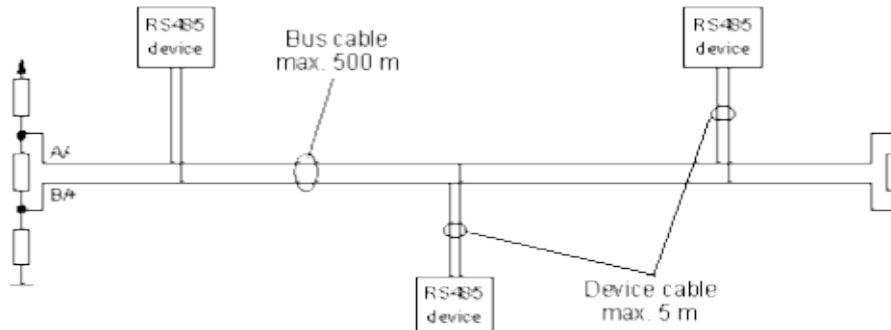
1. Especificaciones funcionales. El Bus RS-485 puede instalarse como un sistema de 2 hilos o como uno de 4 hilos.

Bus de 2 hilos: El estándar 2D-RS-485, es el más comúnmente conocido por estándar RS-485. En este caso se utiliza una sola línea de transmisión balanceada bidireccional. Las características físicas de la línea se mantienen (longitud y velocidades de transmisión admisibles). La diferencia con el bus de 4 hilos es que los dispositivos deben conmutar entre modo receptor y modo transmisor, para evitar que varios dispositivos emitan simultáneamente. (13)

El Bus de 2 hilos tiene una longitud máxima de 500m. Los participantes se conectan a este cable a través de una línea extra de 5 metros de largo como máximo. La ventaja de la técnica de 2 hilos reside esencialmente en la capacidad multimaster, en donde cualquier participante puede cambiar datos en principio con cualquier otro. El Bus de 2 hilos únicamente nos permite establecer una comunicación *half duplex*. Es decir, puesto que sólo hay a disposición una vía de transmisión, siempre puede enviar datos un solo participante. Sólo después de finalizar el envío, pueden responder otros participantes. La aplicación más

conocida basada en la técnica de 2 hilos es el *PROFIBUS*. (8)

Figura 3. Bus RS-485 de 2 hilos (8)



Las señales necesarias en este caso son:

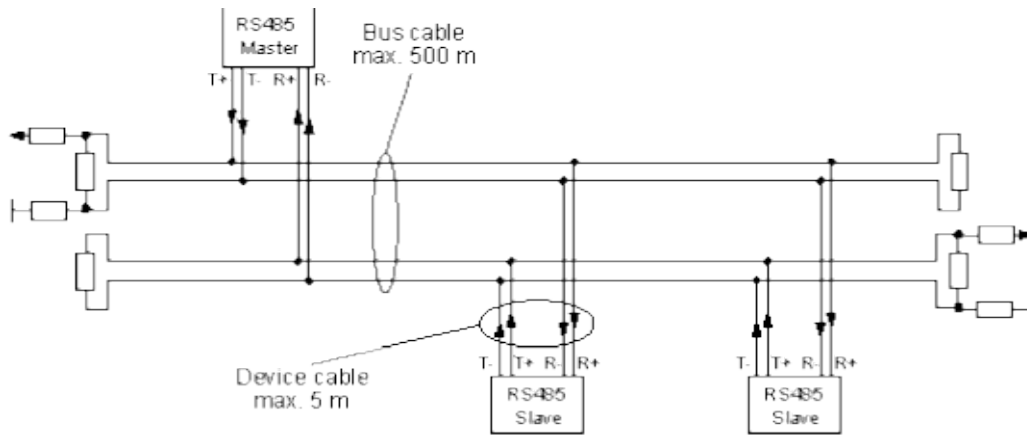
Tabla 2. Señales bus RS-485 de 2 hilos

Nombre	Función
TXD/RXD (+)	Transmisión de datos (Salida +)
TXD/RXD (-)	Transmisión de datos (Salida -)
GND	Tierra

Bus de 4 hilos: Otra de las variantes del estándar RS-485 es la conocida como 4D-RS-485. En este caso se mantienen por separado los pares de cables de recepción y conexión. En este modelo, el equipo principal (maestro), puede mandar datos en cualquier momento, mientras que los equipos secundarios (esclavos), únicamente deben responder cuando el mensaje va dirigido a ellos. (13)

La técnica de 4 hilos puede ser usada por aplicaciones Maestro/Esclavo o full duplex. Conforme al bosquejo se cablea aquí la salida de datos del maestro a las entradas de datos de todos los esclavos. Las salidas de datos de los esclavos están concebidas conjuntamente en la entrada de datos del maestro. (8)

Figura 4. Bus RS-485 de 4 hilos (8)



Las señales necesarias en este caso son:

Tabla 3. Señales bus RS-485 de 4 hilos

Nombre	Función
TXD (+)	Transmisión de datos (Salida +)
TXD (-)	Transmisión de datos (Salida -)
RXD (+)	Recepción de datos (Entrada +)
RXD (-)	Recepción de datos (Entrada -)
GND	Tierra

D. Página web

Una página web es una fuente de información compatible con la World Wide Web (WWW), que puede ser accedida a través de un navegador en internet. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar conexión a otras páginas web mediante enlaces de hipertexto, estos enlaces se conocen como *enlaces externos*. Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan *enlaces internos*. El hipertexto es una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información. (2)

Las páginas web pueden estar almacenadas en un equipo local o un servidor web remoto. Generalmente múltiples páginas web en un mismo dominio constituyen un sitio web. En un sitio pueden existir páginas web de acceso público y de acceso privado. El contenido de la página puede ser predeterminado (página web estática) o generado al momento de visualizarla o solicitarla a un servidor web (página web dinámica). Las páginas dinámicas que se generan, al ser solicitadas, son creadas por una aplicación en el servidor que las contiene. Pero, por lo general, el servidor siempre responderá en un formato HTML. (2)

1. **HTML.** La palabra HTML se deriva de las siglas en inglés HyperText Markup Language y es utilizado para crear la estructura del texto que compone una página web, complementándolo con imágenes y otros contenidos multimedia. Los componentes principales en este tipo de formato son los elementos y sus atributos. Cada elemento posee contenido y atributo. (1)

Para definir un elemento se utilizan etiquetas, las etiquetas son llaves entre las que se encierra el nombre correspondiente del elemento que se desea mostrar, por ejemplo, para mostrar un párrafo se utiliza la etiqueta “<p>” y se termina el párrafo con la etiqueta de cierre “</p>”. Los atributos se escriben dentro de la etiqueta de inicio del elemento, contienen información de cómo se debe mostrar el elemento, como color, tamaño, referencia a otro documento etc.

El tipo de Documento (doctype) es lo primero que debe ir en un documento HTML, no es una etiqueta, es una instrucción que le dice al navegador en qué versión del lenguaje de marcado de hipertexto está escrita la página. Posee una referencia hacia un archivo DTD, Document Type Definition por sus siglas en inglés, en donde están las reglas para renderizar correctamente el contenido de la página. (1)

2. **PHP.** El término PHP viene del acrónimo Hypertext Pre Procesor y es un lenguaje de interpretación que corre del lado del servidor, independiente de la plataforma, o del navegador que está accedendo a la información en el lado del

cliente. El lenguaje PHP nació de la necesidad de crear páginas web dinámicas, y de la posibilidad de controlar qué se muestra y cómo se muestra, no solamente compartir información escrita en un código HTML, sino que permita la interacción con el usuario. (1)

El PHP se utiliza para realizar scripts que permiten una amplia gama de funciones, como por ejemplo accesos a bases de datos, conexiones en red, ciclos, iteraciones y otras tareas dentro de la página. El cliente solamente recibe una página con el código HTML resultante, por lo que es compatible con todos los navegadores. (1)

El lenguaje PHP funciona de la siguiente manera, creado un archivo con extensión php, este debe contener exactamente el mismo código como si se estuviera escribiendo una página en un archivo con extensión html, la única diferencia es que el intérprete buscará las etiquetas especiales “<?php ?>” incrustadas en el contenido HTML para procesarlas en lenguaje PHP. (1)

E. Base de datos

Existen muchas formas de almacenar información, desde crear archivos de texto con los datos deseados, hasta utilizar servidores y estructuras de datos más complejas como las bases de datos. SQL es un lenguaje estandarizado para acceder y manipular bases de datos, está regulado por los estándares ANSI (American National Standards Institute). (5)

El sistema de base de datos que utiliza la torre meteorologica es MySQL. MySQL ofrece ventajas a los otros tipos de bases de datos debido a su alto desempeño, tiene una estructura sencilla y además es gratuito. Es la base de datos más utilizada y debido a su estructura, las consultas de datos resultan rápidas y confiables, solamente se necesita el conector o driver, pero la mayoría de sistemas modernos de programación, como PHP, son totalmente compatibles y no necesitan de ningún driver o controlador adicional. (5)

VI. COMUNICACIÓN SENSOR TORRE

A. Módulo principal o maestro

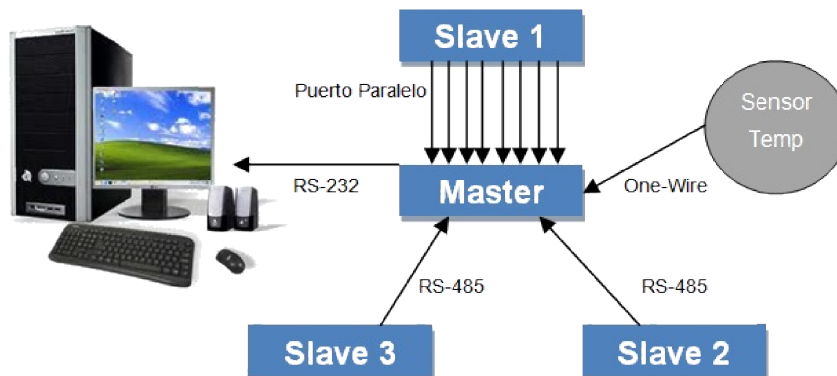
El cerebro de la torre meteorológica es su módulo principal, este posee un microcontrolador Microchip PIC16F887 el cual recibe el nombre de microcontrolador “Master” o Maestro, y otros 3 microcontroladores mas denominados “Slaves” o Esclavos, siendo los mismos Microchip PIC16F887 de igual forma.

Entre las características y funcionalidades del microcontrolador PIC16F887 se pueden encontrar su velocidad de operación de hasta 20Mhz, una memoria de programa (ROM) de 8kbytes con tecnología FLASH, memoria de acceso aleatorio (RAM) de 368 bytes y el módulo USART el cual permite trabajar con los protocolos de comunicación RS-485 Y RS-232. Dicho microcontrolador es presentado con un empaquetado DIP de 40 pines.

Cada microcontrolador esclavo se encarga de almacenar temporalmente la información proveniente del sensor que este mismo tenga a cargo, con el fin de luego transmitírsela al microcontrolador maestro. El maestro tiene la tarea de comunicarse cada minuto con los 3 microcontroladores esclavos y a la vez también comunicarse con el servidor que almacena y publica los datos recolectados en la web.

Para ello, el maestro utiliza distintos protocolos de comunicación adecuándose cada uno al tipo de comunicación que se necesita y a la información que se transmite. A continuación se ilustran los protocolos implementados por el microcontrolador maestro dependiendo de con quién este se comunique. Cabe mencionar que en el diagrama aun no se presenta la implementación del módulo con el sensor de detección de tormentas.

Figura 5. Funciones módulo maestro



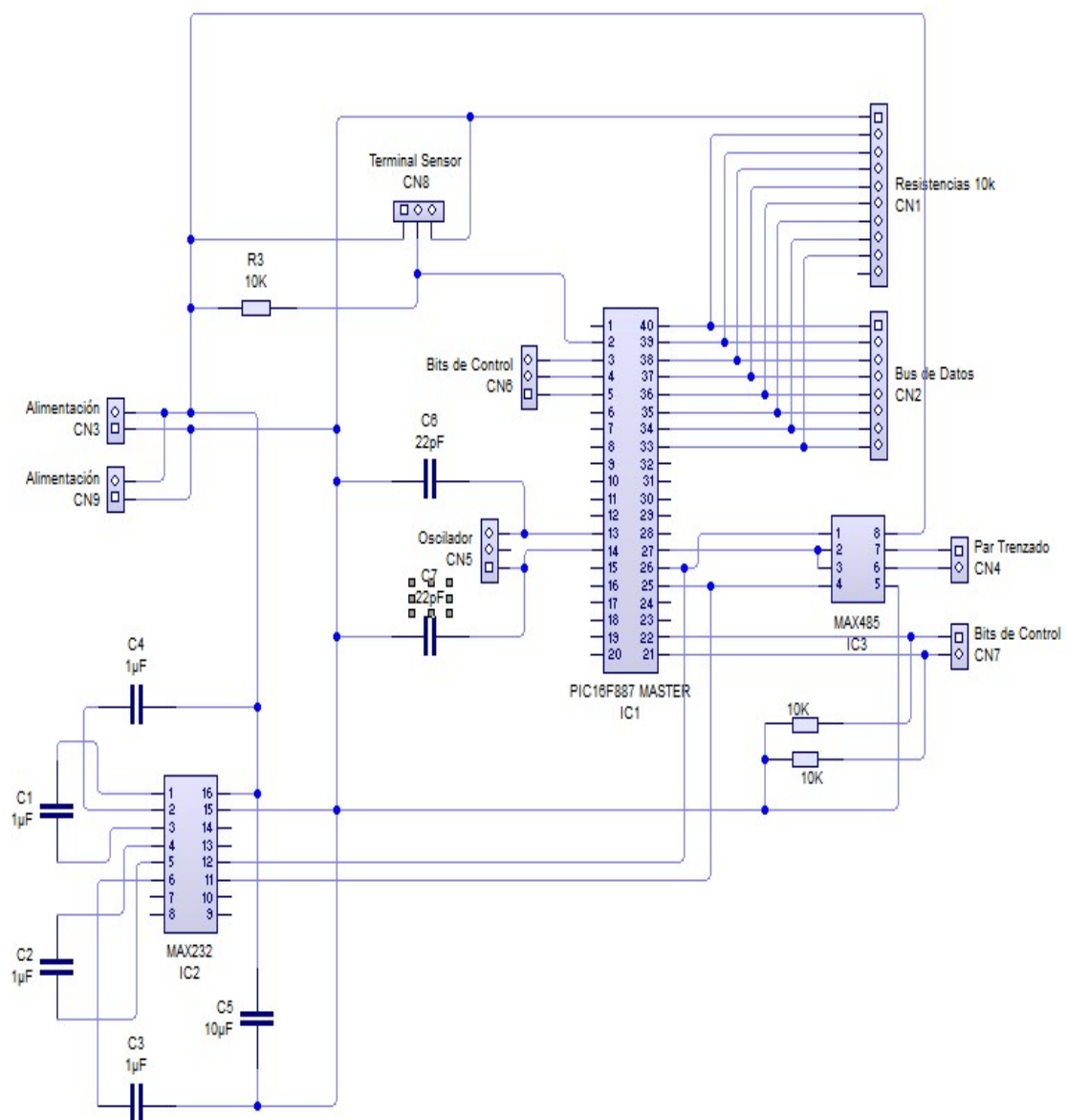
Como se mencionó con anterioridad, el microcontrolador maestro tiene 40 pines. De estos 40 pines, tiene 19 actualmente en uso. La siguiente tabla mostrará cuáles son estos pines y para qué están siendo utilizados.

Tabla 4. Pines utilizados

Pin	Nombre	Función
2	RA0	Protocolo 1-wire (temperatura)
3	RA1	Control Driver de Esclavo 2
4	RA2	Reinicio de Datos Sismos
5	RA3	Control Driver de Esclavo 3
14	RA6	CLKIN (20 MHz)
13	RA7	CLKOUT (20 MHz)
33	RB0	Bus de Datos, línea A0
34	RB1	Bus de Datos, línea A1
35	RB2	Bus de Datos, línea A2
36	RB3	Bus de Datos, línea A3
37	RB4	Bus de Datos, línea A4
38	RB5	Bus de Datos, línea A5
39	RB6	Bus de Datos, línea A6
40	RB7	Bus de Datos, línea A7
25	RC6	Serial Asíncrono, Línea de Transmisión
26	RC7	Serial Asíncrono, Línea de Recepción
21	RD2	Bit de Control Puerto Paralelo 1
22	RD3	Bit de Control Puerto Paralelo 2
27	RD4	Control Driver Maestro

Para poder entender un poco más acerca de la estructura, diseño y funcionamiento del actual sistema de la Torre Meteorológica, se mostrara a continuación una imagen que ilustra el circuito Maestro del sistema. Este diseño nos permite ver claramente la forma en la que opera y gobierna el microcontrolador maestro a todo el sistema. Cabe mencionar que esta imagen aun no muestra la implementación del módulo de detección de tormentas.

Figura 6. Diagrama circuito módulo maestro



Analizando el diagrama presentado, se puede observar cómo se realizó físicamente la implementación de cada protocolo de comunicación. Ya que para poder manejar y utilizar estos protocolos, mencionados con anterioridad, no solamente es necesario trabajar a nivel de software, sino también a nivel de hardware. Tanto para la comunicación en paralelo, como en las comunicaciones seriales RS-232 y RS-485, hay que utilizar componentes electrónicos. Es por eso que dentro del diagrama del circuito se encuentra presentes circuitos integrados o “drivers” como el MAX232 y el MAX485, además de las distintas resistencias y capacitores.

B. Sensor PreviStorm

Como ya se explicó en el rubro: (I) Introducción, (A) Objetivos del Módulo, (1) General del Módulo, el objetivo general o principal de este trabajo es la integración de un nuevo sensor al sistema. Y para ello se necesita conocer también todos los componentes que conforman al sensor, con el fin de obtener el conocimiento de cómo funciona y cómo son proporcionados los datos obtenidos por el mismo.

El sensor de detección de tormentas PreviStorm está conformado por varios componentes, cada uno indispensable para su adecuado funcionamiento. El sensor cuenta con: 1) Fuente de alimentación y protección, 2) Un módulo de adquisición de datos (MAD) y 3) Sensor MCE o medidor de campo electrostático.

La fuente de alimentación y protección permite la continua alimentación para el equipo PreviStorm. Esta fuente se alimenta de 125/220V AC brindándole al sistema del sensor 24V DC. Esta fuente de alimentación y protección es en realidad un UPS dedicado, específico para su aplicación, capaz de dotar al equipo de una alimentación en caso de un fallo en la red general de alimentación, con una autonomía de más de 7 horas.

Figura 7. UPS PreviStorm



El módulo de adquisición de datos MAD (ver Figura 8) es el encargado de brindarle al usuario la información obtenida por el MCE. Este módulo cuenta con una pantalla LCD con iluminación, y 3 indicadores luminosos para las respectivas 3 alarmas que el sistema emite. También cuenta con un teclado de programación, otro indicador luminoso para error o fallo del sistema, y un zumbador el cual se activara juntamente con cualquier alarma.

Figura 8. MAD PreviStorm



Este módulo permite el control y configuración del sistema, el análisis y tratamiento de la señal medida de campo electrostático, la visualización del valor y el disparo de los relés asociados a cada nivel de alarma.

Además de lo anteriormente mencionado, el MAD cuenta con tres puertos. El primero es el de la alimentación, este se conecta a la salida de 24V DC provista por el UPS. El segundo permite la comunicación con el MCE, y finalmente el tercer puerto es un puerto RS-232C, el cual permite que el MAD se pueda conectar a una computadora. Esto con el fin de transmitir los datos adquiridos a una computadora o servidor, y que a partir de esta, la información manejada por el módulo de adquisición de datos pueda ser transmitida a otros equipos por medio de TCP/IP, Modems o una red GSM.

El módulo de adquisición de datos del sensor PreviStorm está mandando constantemente sus datos por medio de su puerto RS-232. El puerto RS-232 está configurado para trabajar con un formato de 8 bits, 1 stop bit y no tiene paridad. Los datos los manda en una cadena codificada que contiene toda la información necesaria de la medición y transmisión. La cadena de 11 bytes de longitud es enviada a una velocidad de transferencia o Baud Rate de 19200 bps, de la siguiente forma:

- a030C050202 -

El sensor del sistema Previstorm o mejor conocido como medidor de campo electrostático (ver Figura 9), es el encargado de obtener el valor del campo electrostático del ambiente y compartirlo con el módulo de adquisición de datos. Este siendo de tipo molino de campo, permite medir de forma dinámica la evolución del campo electrostático, mediante el diferencial de potencial existente entre la nube y el potencial cero de la tierra por unidad de longitud.

Figura 9. MCE PreviStorm



El sistema PreviStorm, es capaz de detectar de forma preventiva el fenómeno del rayo. No es necesaria la caída de un rayo para que detecte algún tipo de actividad tormentosa, y permite en tiempo real tomar decisiones de seguridad ante la acumulación de carga electrostática o descargas electroatmosféricas, garantizando así la seguridad de personas y bienes materiales.

Según sea la medición provista por el sistema PreviStorm, uno puede conocer el estado en el que se encuentra la posible tormenta. Existen ciertos rangos de valores de campo electrostático que nos permiten determinar el desarrollo de la misma, estos se detallan a continuación.

Tabla 5. Estado de la tormenta

Tormenta Lejana	Tormenta Próxima	Tormenta Activa
2kV/m	4kV/m	8kV/m
Formación inicial de la tormenta, baja probabilidad de descargas a tierra.	Actividad entre nubes intensa. Mayor probabilidad de descargas a tierra.	Alta probabilidad de riesgo de caída de rayos a tierra.

C. Establecer comunicación entre el sensor y el módulo maestro

1. **En la práctica.** Una vez comprendida la funcionalidad del modulo principal o Maestro y el sensor, es necesario crear la comunicación entre ambos. Como se mencionó anteriormente, el módulo de adquisición de datos del sensor, comparte las mediciones a través de un puerto RS-232C, por lo que para lograr comunicar al sensor con el microcontrolador maestro es necesaria la utilización de un driver MAX232, otro microcontrolador esclavo, y un driver MAX485.

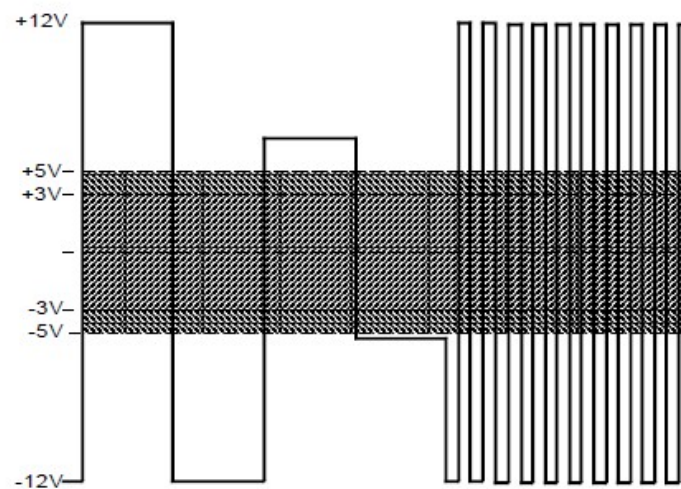
El microcontrolador esclavo debe ser capaz de manejar dos comunicaciones seriales, con configuración distinta, al mismo tiempo. Debe notarse que la velocidad de transmisión o *Baud Rate* con la que opera el puerto serial del MAD del sistema de detección de tormentas es distinta a la velocidad de transmisión y recepción de datos del modulo principal de la torre meteorológica. Los equipos dentro de la torre meteorológica que se comunican de forma serial, lo hacen a una velocidad de operación de 4800 bps. Por esa razón se está utilizando el microcontrolador dsPIC30F4011, para lograr la realización de dicha tarea.

Entre las características y funcionalidades del dsPIC30F4011 se puede encontrar que es un microcontrolador de señales digitales de 16 bits, con una velocidad de operación de hasta 40Mhz, una memoria de programa (ROM) de 48kbytes con tecnología FLASH, memoria de acceso aleatorio (RAM) de 1 kbyte y

doble módulo UART el cual permite trabajar con dos protocolos de comunicación RS-232. Dicho microcontrolador es presentado con un empaquetado DIP de 40 pines.

Para establecer la comunicación del MAD con el esclavo, es necesario un driver MAX232, este driver permite convertir la señal de salida del RS-232C, la cual se encuentra en el rango de $\pm 12V$ DC, a una señal con un nivel de voltaje apropiado para dispositivos TTL o CMOS, los cuales no pueden tener un nivel de voltaje mayor a +5V DC en sus terminales.

Figura 10. Niveles eléctricos RS-232 vs. dispositivos TTL o CMOS



El microcontrolador maestro del módulo principal utiliza el protocolo RS-485 para manejar a dos de sus microcontroladores esclavos. Por medio del driver MAX485 el maestro puede indicarles a los esclavos cuando quiere recibir la información que estos le están enviando constantemente. Es por esta razón que es necesario el uso de otro driver MAX485 para que el maestro pueda manejar al nuevo microcontrolador esclavo del sistema de detección de tormentas.

El driver MAX485, o su equivalente SN65176B, opera mediante tres modos distintos de operación, los cuales se pueden seleccionar según la señales que

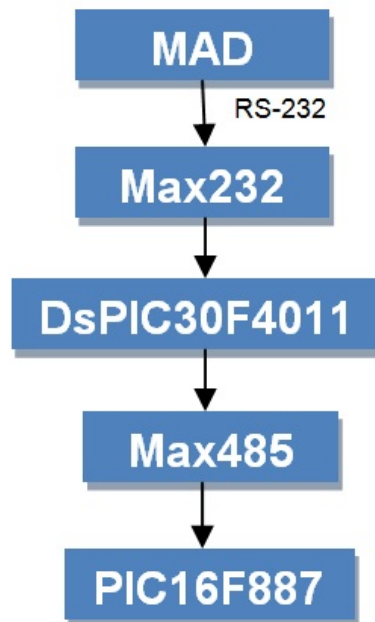
estos tengan en sus terminales –RE y DE. En la siguiente tabla se mostrara cuales deben de ser las señales en esos pines del driver para seleccionar su estado de transmisión, recepción o neutro (apagado).

Tabla 6. Estados driver MAX485

-RE	DE	Estado
1	1	Transmite
0	0	Recibe
1	0	Neutro (Apagado)

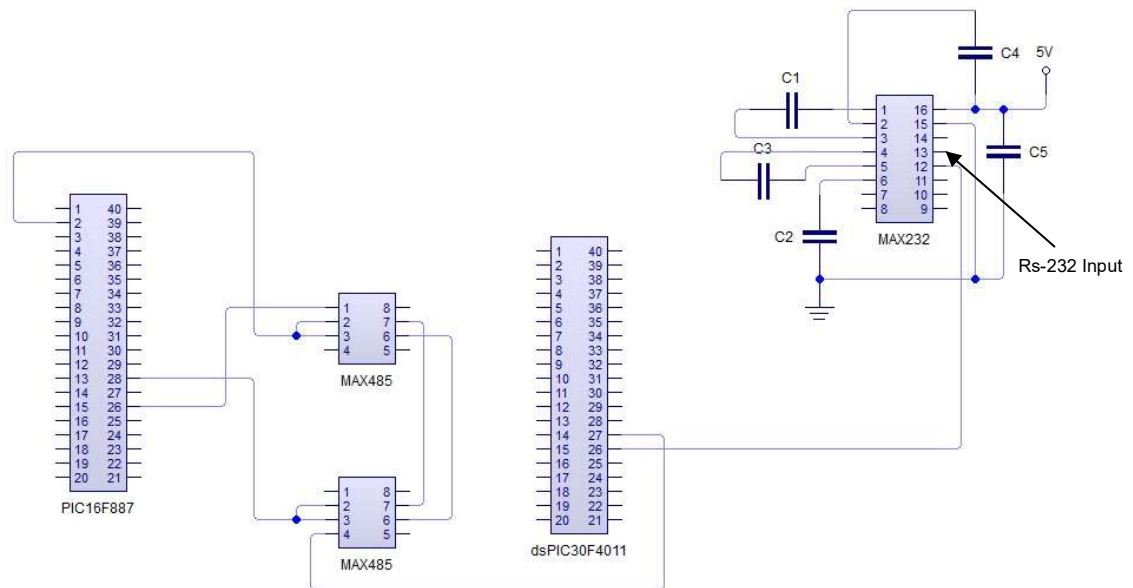
¿Quién decide cuándo va a transmitir información cualquiera de los esclavos? Esa es parte de la responsabilidad del microcontrolador maestro, ya que él es el que le solicita los datos a los esclavos, cuando sean necesarios, para transmitirlos a la base de datos del servidor. Cabe mencionar que el maestro solo recibe datos, no envía nada a los microcontroladores esclavos. De esta forma, los esclavos envían sus datos, pero no reciben nada del maestro.

Gracias a la implementación del driver MAX485, el esclavo pueden estar mandando siempre los datos provistos por el MAD aunque no sean escuchados por el maestro. En este tipo de comunicación, el driver MAX485 del maestro siempre está en modo escuchar, recibiendo todos los datos. El driver MAX 485 del módulo de detección de tormentas se encuentran en estado neutro, pero siempre están mandando sus datos. Cuando el maestro quiere el dato de alguno de los esclavos, o en este caso del esclavo que recibe la información del MAD del sensor PreviStorm, solo se cambia el estado del driver MAX485 que se encuentra en este circuito, de neutro a transmitir, de igual forma que lo hace con los otros microcontroladores esclavos.

Figura 11. Comunicación maestro - MAD

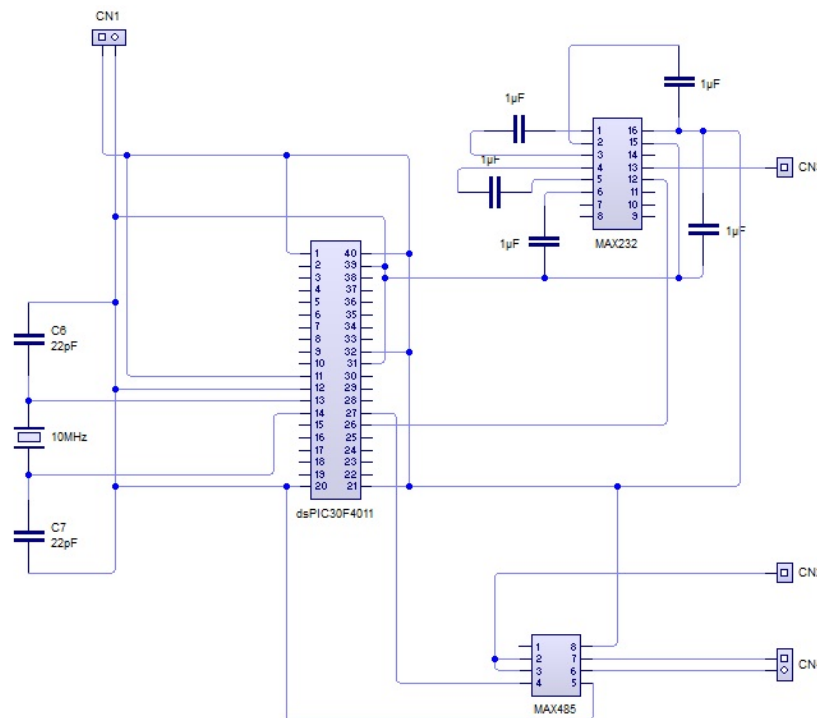
Debido a que las únicas ordenes que nos interesan darle al driver MAX485 son las de transmitir y recibir, realizamos un corto circuito entre -RE y DE , con el objetivo de que cuando el maestro envíe un 0, el driver configure su estado a recibir y cuando envíe 1 se configure para transmitir. A continuación se mostrará un gráfico sencillo ilustrando la conexión a utilizar entre el microcontrolador esclavo dsPIC30F4011, con los drivers MAX232 y MAX485, con el MAD y el microcontrolador maestro:

Figura 12. Integración maestro - esclavo



Partiendo de este diseño inicial se procedió a realizar una versión formal y funcional, con el objetivo de poder diseñar y construir el circuito del nuevo módulo de detección de tormentas. La siguiente imagen ilustra la versión inicial del circuito del módulo de detección de tormentas.

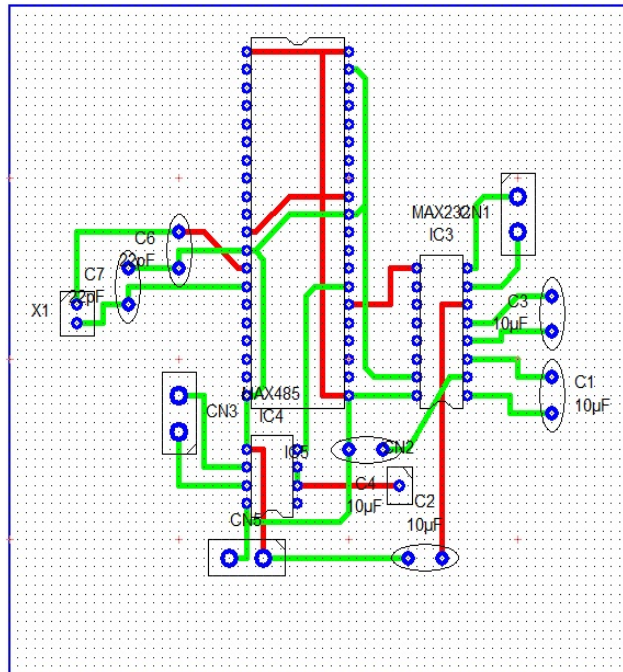
Figura 13. Circuito módulo de detección de tormentas (Versión inicial)



Teniendo listo el diseño del circuito necesario para lograr la implementación del módulo de detección de tormentas, se procedió a la creación física del mismo. Por lo que, utilizando como base el diseño de la Figura 13, se creó la placa del circuito.

Así mismo se creó el programa con las instrucciones a seguir por el microcontrolador esclavo dsPIC30F4011. Dicho programa era sencillo, el microcontrolador esclavo únicamente estará guardando en su memoria la información recibida de forma serial proveniente del MAD, una cadena de 11 Bytes. Una vez almacenados los datos en su memoria interna, el microcontrolador esclavo empezará a transmitir por su otro puerto serial y a una menor velocidad dicha información. Al finalizar la transmisión, el programa vuelve a repetirse desde el inicio, de manera cíclica.

Figura 14. Placa circuito



En la imagen mostrada, se puede observar la primera versión del circuito. Con dicho diseño impreso en una placa de cobre y con cada componente electrónico en su lugar, se procedió a realizar las primeras pruebas con el fin de verificar el adecuado funcionamiento del mismo. Ya que este fue creado a partir de toda la información teórica recaudada y descrita anteriormente en este trabajo.

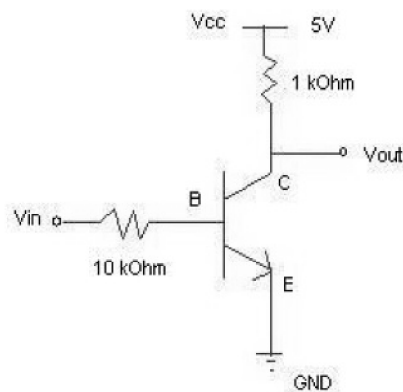
Para poder analizar el funcionamiento del circuito fabricado, se utilizó un programa de hiperterminal para poder ver los datos enviados de forma serial desde el sensor y el microcontrolador esclavo dsPIC30F4011 dentro de dicho circuito. Se pudo observar que el sensor PreViStorm estaba transmitiendo sus datos sin ningún problema. Sin embargo, el microcontrolador esclavo no estaba recibiendo bien estos datos, ya que al observar los datos guardados y transmitidos por el mismo, estos no coincidían con los datos que el mismo estaba recibiendo.

Luego de revisar que tanto el microcontrolador esclavo dsPIC30F4011, como el driver MAX232 estuvieran funcionando y realizando sus tareas de manera

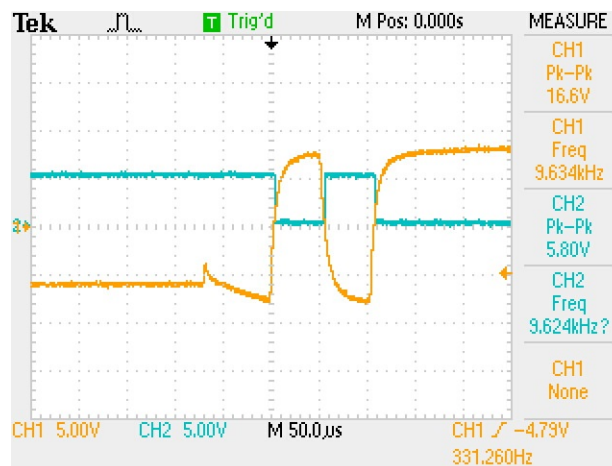
correcta, se llegó a la conclusión de que el problema estaba relacionado con las impedancias y corrientes eléctricas dentro del circuito. Por lo que se busco sustituir el driver MAX232 por otros componentes o circuitos que realizaran exactamente la misma tarea, pero eliminando dicho problema.

El siguiente paso fue implementar un circuito inversor de voltaje utilizando un transistor NPN con el fin de sustituir al driver MAX232. Este es un circuito sencillo conformado por dos resistencias y un transistor 2N3904, alimentado con 5V DC. A continuación se presenta un diagrama del circuito inversor de voltaje.

Figura 15. Inversor de voltaje (Transistor NPN)



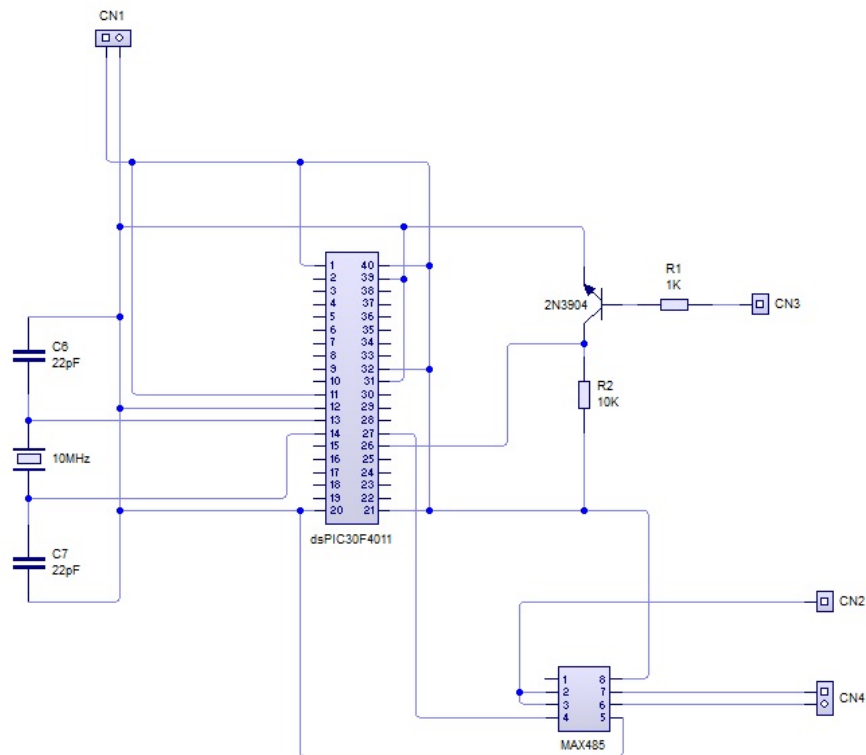
El inversor de voltaje realiza la misma función que el driver MAX232, no sólo invierte la señal que el mismo recibe en $\pm 12V$ DC, sino la reduce a una señal con un nivel de voltaje no mayor a +5V DC en sus terminales. La siguiente imagen es una muestra del funcionamiento real de este circuito.

Figura 16. Voltaje de entrada vs. voltaje de salida (Inversor)

La gráfica naranja es la señal proveniente del puerto serial del MAD, la cual es la señal de entrada (V_{in}) del circuito inversor. La gráfica azul es la salida del circuito inversor de voltaje (V_{out}). Al analizar la imagen, se puede ver claramente que el circuito inversor de voltaje está cumpliendo con su función.

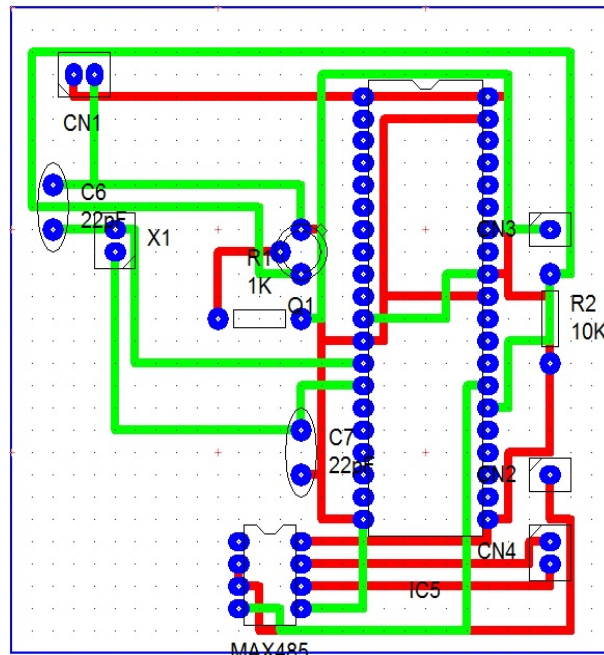
Comprobado el funcionamiento del circuito inversor, se repitió el procedimiento inicial. Primero se volvió a realizar el esquema del nuevo circuito con el cambio ya implementado, dando como resultado el circuito mostrado a continuación.

Figura 17. Circuito módulo de detección de tormentas (Versión mejorada)



Con este nuevo esquema, se procedió a realizar la versión del circuito en una placa de cobre, al igual que la vez anterior. Se reordenaron los componentes dentro de la placa y se llegó al resultado final mostrado en la siguiente figura.

Figura 18. Placa circuito (Versión mejorada)



Ya con el circuito y cada componente electrónico listo y montado en la placa de cobre, se volvió a probar nuevamente. Esta vez los datos recibidos, almacenados y reenviados nuevamente por el microcontrolador esclavo dsPIC30F4011 eran los mismos provenientes del puerto serial del MAD. Una vez pasada esta prueba, se concluyó que el circuito presentado en la Figura 18 sería la versión final del circuito utilizado para la implementación del nuevo módulo de detección de tormentas dentro del sistema de la Torre Meteorológica.

Teniendo listo el hardware de control de transmisión de datos del sensor de detección de tormentas, se procedió a modificar el programa dentro del microcontrolador maestro. Se asignó al mismo el control de un driver MAX485 más, así como la recepción de una nueva cadena de datos. La nueva variable de campo electrostático debe de ser almacenada temporalmente dentro del listado de variables climáticas dentro de la memoria del maestro.

Todas las variables que son medidas por otros sensores son, de igual forma, almacenadas temporalmente y enviadas a la computadora en forma de

cadena de datos, por lo que se debe de enviar una nueva cadena con el valor de campo electrostático. Esta cadena se identifica con un encabezado de la siguiente forma:

Cadena "\$WIMDC" (Campo Electroestático)

Seguido de los 11 bytes provenientes del MAD del sensor de detección de tormentas, teniendo como resultado final la cadena mostrada a continuación:

\$WIMDC,a,0,2,2,6,0,1,0,0,1,4,

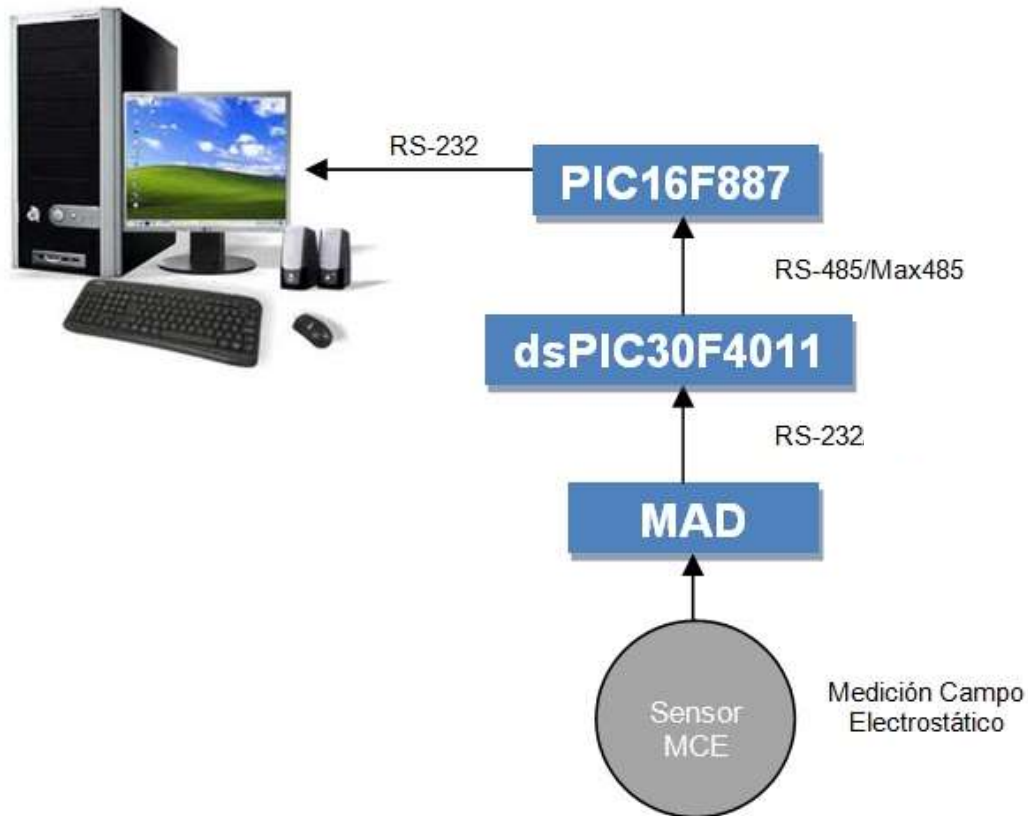
La nueva cadena es recibida por el servidor y almacenada en la base de datos del sistema de la torre meteorológica. Se transmitirá de esta forma para seguir el mismo formato establecido por la Torre Meteorológica.

Como se mencionó con anterioridad, el módulo principal se comunica por medio de RS-232 con el servidor que contiene en su base de datos todas las variables climáticas medidas, siendo esta información transmitida en las siguientes cadenas, todas enviadas por el maestro. Estas contienen toda la información necesaria para su posterior almacenamiento y clasificación en la base de datos del servidor.

- Cadena "\$WIMDA" (Dirección y velocidad del viento, Humedad Relativa y presión barométrica)
- Cadena "\$WIMDE" (Temperatura)
- Cadena "\$WIMDI" (Sismos)
- Cadena "\$WIMDO" (Radiación)
- Cadena "\$WIMDC" (Campo Electroestático)
- Cadena "\$WIMDU" (Número de Torre Meteorológica)

La comunicación del módulo de adquisición de datos Previstorm con el microcontrolador maestro y su función se ilustrara a continuación:

Figura 19. Funciones módulo detección de tormentas



Finalmente se comprobó, por medio de la hiperterminal, que el microcontrolador maestro estaba recibiendo, guardando y transmitiendo vía serial los datos correctos provenientes del MAD. Siendo el microcontrolador esclavo dsPIC30F4011 el “puente” por el cual los mismos eran transmitidos del MAD al microcontrolador maestro.

VII. COMUNICACIÓN TORRE SERVIDOR

Para que los datos recopilados por la torre meteorológica puedan ser estudiados y analizados, es necesario contar con una base de datos y para ello es indispensable el uso del servidor. El servidor está destinado para almacenar todos los datos provistos por la torre meteorológica en una base de datos que se encuentra instalada en el mismo. Las características del servidor utilizado por la Torre Meteorológica y el nuevo módulo de detección de tormentas se muestran a continuación:

Tabla 7. Características del servidor

Modelo del servidor	DELL PowerEdge 2650
Sistema operativo	Windows Server 2008
Capacidad de almacenamiento	250 GB
Memoria RAM	1 GB DDR1

Para que el módulo de comunicación y página web funcione adecuadamente, el servidor debe estar en línea todo el día, todos los días, de lo contrario, no se podría contar con una base de datos robusta para realizar estudios climáticos. La base de datos dentro del servidor es el lugar en donde se encuentra almacenados todos los datos recopilados por la torre meteorológica. Los datos se transmiten al servidor por medio de un software, el cual es una herramienta importante para la constante actualización de la base de datos.

A. Software de comunicación

El software de comunicación es una herramienta creada en lenguaje de programación de alto nivel C# y se encuentra instalado dentro del servidor. Este software se encarga de decodificar cada una de las cadenas "\$WIMDx" provenientes de la torre meteorológica, para luego poder almacenar los datos que vienen dentro de las mismas con su valor y dimensión correspondiente dentro de la base de datos del servidor. Para la cadena con encabezado "\$WIMDC" proveniente del módulo de detección de tormentas, es necesario utilizar una

fórmula provista por INGESCO (empresa que diseña, construye y manufactura el sensor de detección de tormentas utilizado).

Como se menciona anteriormente en el rubro (V) COMUNICACIÓN SENSOR TORRE (B) Sensor Previstorm, el MAD envía una cadena de 11 bytes y a continuación se describirá que es lo que cada uno de estos representa.

Tabla 8. Cadena – Trama de envío

Byte	Información
1	Encabezado cadena ('a')
2 a 5	Valor del campo 16 bits 256 = 1kV/m y 25600 = 100kV/m
6 y 7	Tipo de medición Bit 0-1 = Medición Bit 2 = Signo
8 y 9	Estado Bit 0-1 = Alarma 0 No alarma 1 Alarma 1 2 Alarma 2 3 Alarma 3 Bit 2 = 1 Medida Bit 3 = 1 Motor Bit 4 = 1 Temperatura Bit 5 = 1 Comunicación Bit 6-7 = Control
10 y 11	Complemento a dos de la suma de 8 bits del byte 1 al 9

Una vez entendido lo que cada byte dentro de esta cadena representa y el algoritmo utilizado para decodificar dicha cadena de 11 bytes. Se modificó el software de comunicación, agregándole al mismo el algoritmo necesario para la

lectura y decodificación de la nueva cadena de datos proveniente del módulo de detección de tormentas. Se desecharon los encabezados de identificación y se utilizaron únicamente los bytes del 2 al 7 para obtener el campo electrostático representado en un valor numérico con signo y guardar esta información en la base de datos del sistema Torre Meteorológica. Tampoco se utilizaron los bytes del 8 al 11 debido al fin y delimitación del proyecto.

Figura 20. Software de comunicación



B. Base de datos

La base de datos con la que cuenta la torre meteorológica, fue estructurada a partir de dos tablas. La primera tabla es la de las mediciones, esta contiene toda la información que ha sido recopilada a lo largo del tiempo por el sistema. En ella se encuentran los valores numéricos de todas las variables climáticas recolectadas, un índice correlativo para llevar el control de la cantidad de datos que se han almacenado, y campos de información de fecha y hora para establecer el momento exacto en el que se muestrearon los datos correspondientes. Por lo que para almacenar la nueva variable de campo electrostático dicha tabla de la base de datos fue modificada. En la Tabla 3. Estructura de la Tabla Mediciones, se muestra el orden de los datos en la nueva tabla.

Tabla 9. Estructura de la tabla mediciones

Numero	Columna	Tipo de dato
1	correlativo	Int, Auto-Increment
2	Id_estacion	Text
3	temperatura	Float
4	viento_velocidad	Float
5	Viento_direccion	Float
6	presión	Float
7	humedad_relativa	Float
8	punto_rocio	Float
9	radiacion	Float
10	Sismo	Float
11	campo_electrostatico	Float
12	fecha	Date
13	hora	Time

La segunda tabla contenida en la base de datos es la tabla de control de errores, la cual es útil para almacenar cualquier tipo de error que pueda ocurrir durante la comunicación entre el módulo de comunicación y pagina web, y el módulo principal. Debido al alcance del trabajo de implementación de otro sensor a la torre meteorológica, y a que esta tabla no fue creada para almacenar errores de comunicación entre los microcontroladores esclavos y el maestro, no fue necesario modificar dicha tabla. Los errores almacenados en la misma únicamente son de fallo de comunicación entre el módulo de comunicación serial RS232 de la torre meteorológica como tal y el servidor.

Una vez reconfigurado el servidor y reinicializada la conexión, el sistema actualizará los datos cada vez que la torre meteorológica se los envíe y el software de comunicación los interprete. Al haber logrado la conexión, se genera el Script para la actualización del contenido en la nueva base de datos, se realiza un Query que obtiene la información recibida y posteriormente se ejecuta para actualizar los datos.

La información almacenada en la base de datos puede consultarse solamente desde la página web del proyecto, los datos de las variables climáticas son de acceso público, mientras que la información de errores solo está disponible para los administradores del servidor.

C. Página web

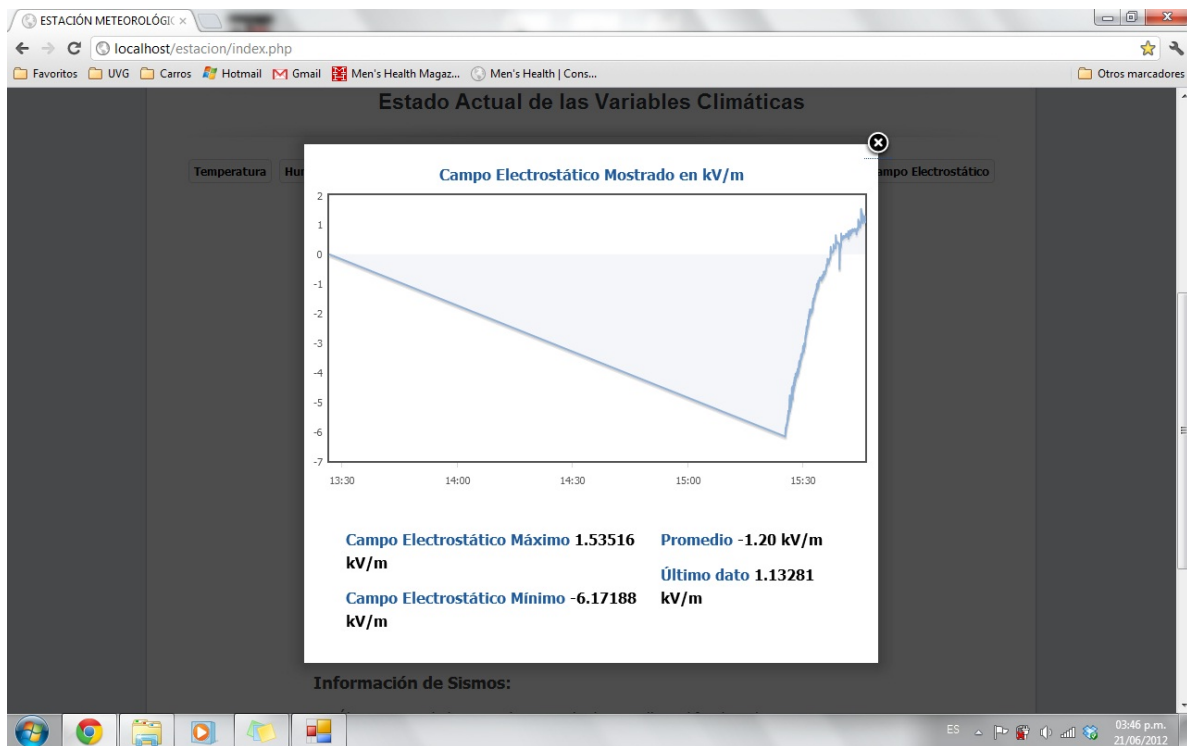
Para que la nueva variable climática pueda ser observada en la página web, se modificó la misma también, de lo contrario aunque la base de datos del servidor la almacene, esta no podrá ser vista desde la página del proyecto.

La ventana principal de la página web de la torre meteorológica está escrita en lenguaje PHP con el fin de actualizar el contenido más reciente almacenando en la base de datos del servidor. En esta ventana se despliega la medición más reciente de cada variable, por lo que se le agrego a la misma el despliegue del valor del campo electrostático.

Para cada variable climática, existe una ventana individual, haciendo necesaria la existencia de una ventana individual para la nueva variable de campo electrostático. La ventana individual del campo electrostático muestra el último dato muestreado por la torre meteorológica y un promedio de los valores obtenidos a lo largo del día. De igual forma que las demás variables, la variable de campo electrostático cuenta con un pequeño informe o historial, el cual muestra su comportamiento a lo largo del día. El usuario puede acceder al mismo al hacer clic en el ícono representativo ubicado en la parte inferior de la ventana individual de la variable.

De esta forma se abre una ventana emergente, la cual contiene una gráfica ilustrando las variaciones de la variable a lo largo del día, al igual que su valor máximo y mínimo. Este pequeño informe sólo muestra la información recolectada el día en el que se está visitando la página, pero también se puede acceder a un historial completo que contiene todos los datos recopilados desde el inicio.

Figura 21. Ventana emergente campo electrostático



La página web también muestra un mensaje indicando la fecha y hora en que se realizó la última actualización de datos y se encuentra localizada en la parte inferior. Cuando la página fue creada, se configuró para actualizar su contenido cada minuto. Y cuando no se obtienen datos durante el día, debido a algún problema como la pérdida de conexión con la torre o el servidor, se muestra un mensaje indicando que no hay datos disponibles.

D. Historial

Como se mencionó anteriormente, existe un historial completo para cada variable climática, en este se encuentra registrados todos los datos almacenados en la base de datos del servidor, desde el primero hasta el más reciente. Este puede ser accesado desde el menú principal superior de la página web.

Aquí, se muestra la gráfica de comportamiento de todas las variables, dando al usuario la opción de elegir cuál de todas las variables desea observar,

entre ellas ahora se encuentra la de campo electrostático. Así mismo, el usuario puede elegir un rango de fechas de interés para realizar la búsqueda de datos de la variable. Estas casillas de selección se encuentran en la parte inferior de la página, al igual que la tabla de datos correspondiente a la información que se está graficando.

Como con el resto de variables climáticas, también es posible descargar la información de interés de la variable de campo electrostático utilizando el botón de descarga ubicado en la parte superior derecha de la tabla.

E. Complementos

La página web también cuenta con una serie de pestañas con información complementaria acerca del proyecto completo de la torre meteorológica. Estas pestañas son puramente informativas y ninguna contiene información de las variables climáticas. Una de las pestañas de interés es la de “Acerca de”, ya que en esta se encuentra una breve descripción del proyecto, además de información de los alumnos involucrados y las actividades realizadas por los mismos dentro del proyecto.

Dentro de la pestaña o sección “Imágenes del Proyecto”, ahora se pueden encontrar imágenes del nuevo sensor de detección de tormentas PreviStorm. Además de las ya contenidas con anterioridad de la estructura, el panel solar y los circuitos impresos, entre otras.

Finalmente en la pestaña “Quiénes somos”, se da crédito a los cinco integrantes del trabajo de investigación inicial y a los estudiantes que han seguido desarrollando, hasta la fecha, aportes de crecimiento para el proyecto. También se encuentran los curriculum vitae de cada alumno que ha formado parte de dicho proyecto para que puedan ser descargados por las empresas o personas interesadas.

VIII. CONCLUSIONES

1. Se diseñó e implementó un módulo capaz de brindar información útil para la predicción de tormentas mediante la medición del campo electrostático.
2. Se agregó la nueva variable de campo electrostático al sistema de la Torre Meteorológica, permitiendo de esta forma estudiar más a profundidad el estado climático.
3. Se logró realizar de manera exitosa la comunicación del sensor de detección de tormentas con el servidor.
4. La conversión de la información provista por el sensor a la variable climática de interés, campo electrostático, se realiza exitosamente.
5. Las mediciones de campo electrostático son almacenadas exitosamente en el servidor.
6. Se tiene acceso a las mediciones del campo electrostático recopiladas por el sensor desde cualquier lugar y en cualquier momento.

IX. RECOMENDACIONES

1. Se recomienda darle mantenimiento a la torre meteorológica, ya que se debe de remover el polvo que va a parar a la misma, así como engrasar los mecanismos, y verificar el continuo funcionamiento de cada uno de sus componentes.
2. Para proyectos como estos, se recomienda entregar un manual donde se detalle la puesta en marcha del sistema completo, así como los pasos a seguir para su desconexión y apagado.
3. De igual forma, se recomienda solicitar y guardar los archivos utilizados para la creación de las placas de los circuitos dentro del sistema, así como el código fuente empleado en cada módulo, con el fin de que otros estudiantes puedan contar con ellos a la hora de ser necesaria una mejora o modificación.
4. Se recomienda fomentar el continuo desarrollo e investigación del proyecto de la torre meteorológica, con el objetivo de poder mejorar su diseño y obtener una mejor herramienta para el estudio climático con el paso del tiempo.

X. BIBLIOGRAFÍA

1. Cordón, J., Cruz, J., Estrada, J., Estrada, F., & Mérida, L. *Megaproyecto Estación Meteorológica*. 2011, Tesis, Universidad del Valle, Guatemala.
2. Díaz, P., Catenazzi, N. Aedo, I. (1996): "De la Multimedia a la Hipermedia". RA-MA Editores, Madrid. 1996.
3. Fernández, L. M. *Protocolo RS-232*. Málaga, Esp Electronic Design, [En línea] http://www.espelectronicdesign.com/Protocolos/protocolo_RS232.php.
4. Feynman, R., Leighton, R., & Sands, M. *The Feynman Lectures on Physics*. 1964, Estados Unidos, Addison-Wesley, Volumen 2, 518 páginas.
5. *Introducción a SQL y SQL Server*. [En línea] http://www.w3schools.com/sql/sql_intro.asp.
6. Jiménez, M. *Rs 232*. [En línea] <http://rdedatos.tripod.com/rs232.htm>.
7. Quintela, F., & Redondo, R. *Campo Electrostatico*. Universidad de Salamanca, 2007, España. [En línea] http://campus.usal.es/~electricidad/Principal/Fenomenos/Publicaciones/Descargas/01_Campo_electrost.pdf
8. LedMatic. *Protocolo RS485*. 2010, España. [En línea] <http://www.pantallas-electronicas.es/index.php/es/maes-lejos-con-rs485.html>.
9. *Localización de Rayos en Áreas*. España, Ingesco. [En línea] http://www.ingesco.com/images/stories/products/FT_PREVISTORM_NET_LLA_esp.pdf
10. *Manual de Usuario*. Edición 2002, España, PreviStorm, 38 páginas.

11. *Módulo Avisador de Caída de Rayos*. España, Ingesco. [En línea] http://www.ingesco.com/images/stories/products/FT_PREVISTORM_NET_LW_esp.pdf
12. Morales, O. E. *Interfase Graficadora para el Medidor de Nivel*. Diciembre 2003, Tesis, Universidad de las Américas Puebla, México. [En línea] http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/morales_h_oe/capitulo3.pdf.
13. *RS485/EIA-485*. Tecnólogos en Mantenimiento Electrónico e Instrumental Industrial. [En línea] <http://tecnologiaelectronik.blogspot.com/>.
14. Sandoval, J. (2009). *Tropic Hardware y Electrónica*. Octubre 2011, EUSKALNET. [En línea] <http://www.euskalnet.net/shizuka/rs232.htm>.
15. Soltero, M., Zhang, J., & Cockril, C. *RS-422 and RS-485 Standards Overview and System Configurations*. Mayo 2010, Estados Unidos, Texas Instruments, 25 páginas.
16. Zozaya, A. *Campo Electrostático*. Noviembre 2009. Facultad de Ingeniería, Universidad de Carabobo. [En línea] <http://www.ing.uc.edu.ve/~azozaya/docs/tem1/CE.pdf>

XI. ANEXOS

A. Anexo 1 – Código Fuente Microcontroladores

1. Código Fuente Módulo Principal

```
//UNIVERSIDAD DEL VALLE DE GUATEMALA
//PROGRAMA PIC 16F887
//MODULO PRINCIPAL

char rxchar, k=0;
int cont3=1;
unsigned char rxarray[80];
char i = 0;
char j = 0;
char cont1=0, flag=0, cont2=0, flag1=0, flag2=0, flag3=0;

const unsigned short TEMP_RESOLUTION = 12;
char tempch[7] = "000.00";
unsigned temp5;

void interrupt ()
{
flag3=1;
    if (PIR1.RCIF)
    {
        rxchar = UART1_Read();

        if (flag1==1)
        {
            rxarray[k] = rxchar;
            K++;
            if (rxchar=='$' && k>5)
            {
                flag=1;
                flag1=0;
                flag3=0;
                PORTD.F4=0;
                PORTD.F2=0;
                PORTA.F3=0;
            }
        }
        if (rxchar == 10)
        {
            flag1=1;
        }
    }
}

void GetTemp(unsigned int temp2write)
{
```

```

const unsigned short RES_SHIFT = TEMP_RESOLUTION - 8;
char temp_whole;
unsigned int temp_fraction;

// Chequear si la Temperatura es Negativa
if (temp2write & 0x8000) {
    tempch[0] = '-';
    temp2write = ~temp2write + 1;
}

// Obtener la parte entera de la temperatura
temp_whole = temp2write >> RES_SHIFT ;

// Convertirlo a caracteres
if (temp_whole/100)
    tempch[0] = temp_whole/100 + 48;
else
    tempch[0] = '0';

tempch[1] = (temp_whole/10)%10 + 48;           // Obtener la decena
tempch[2] = temp_whole%10 + 48;             // Obtener la unidad

// Obtener la parte decimal de la temperatura
temp_fraction = temp2write << (4-RES_SHIFT);
temp_fraction &= 0x000F;
temp_fraction *= 625;

// Convertir la parte decimal a caracteres
tempch[4] = temp_fraction/1000 + 48;       // Obtener los miles
tempch[5] = (temp_fraction/100)%10 + 48;   // Obtener la centena
}

void main()
{
    TRISD = 0x00;           // set direction
to be output
    PORTD = 0x00;          // turn OFF the
PORTD leds
    TRISB = 0xFF;         // set direction
to be output
    PORTB = 0x00;
    ANSEL = 0;             // pone e
    ANSELH = 0;
    TRISA = 0B00110111;
    PORTA =0x00;
    C1ON_bit = 0;         // Disable comparators
    C2ON_bit = 0;
    PORTD.F4=0;
    PORTA.F3=0;
    UART1_Init(4800);
    PIE1.RCIE = 1;
    INTCON.PEIE = 1;
    INTCON.GIE = 1;      // Initialize UART module at 9600
bps
    Delay_ms(100);
    UART1_Write_Text("TORRE METEOROLOGICA");
}

```

```

do
{
    //UART1_Write('I');
    if (PORTA.F1==1 && cont3==1 && PORTA.F2==0)
    {

        cont3=2;
        rxarray[i]=PORTB;
        i++;
        //UART1_Write('J');

    }

    if (PORTA.F2==1 && cont3==2 && PORTA.F1==0)
    {

        cont3=1;
        rxarray[i]=PORTB;
        i++;
        //UART1_Write('K');

    }

    if (rxarray[i-1]=='$')
    {
        cont1++;
        if (cont1==2 && i>20)
        {
            // UART1_Write('Y');
            Ow_Reset(&PORTA, 0);           // Onewire señal
de reseteo
            Ow_Write(&PORTA, 0, 0xCC);     // Comando
SKIP_ROM
            Ow_Write(&PORTA, 0, 0x44);     // Comando
CONVERT_T
            Delay_us(120);

            Ow_Reset(&PORTA, 0);
            Ow_Write(&PORTA, 0, 0xCC);     // Comando
SKIP_ROM
            Ow_Write(&PORTA, 0, 0xBE);     //Comando
READ_SCRATCHPAD

            temp5 = Ow_Read(&PORTA, 0);    //Leer los
primeros 8 bits
            temp5 = (Ow_Read(&PORTA, 0) << 8) + temp5;
//Leer los ultimos 4 bits

            GetTemp(temp5);                //Lllamar al metodo en
este momento                             //se
tiene la temperatura en arreglo tempch

            Delay_ms(100);

```

```

        j=0;
while (j<i)
{
    UART1_Write(rxarray[j]);
    j++;
}
        j=0;

while (j<81)
{
    rxarray[j]='0';
    j++;
}

j=0;
    rxarray[0]='$';
    rxarray[1]='W';
    rxarray[2]='I';
    rxarray[3]='M';
    rxarray[4]='D';
    rxarray[5]='E';
    rxarray[6]=',';
    rxarray[7]= tempch[0];
    rxarray[8]= tempch[1];
    rxarray[9]= tempch[2];
    rxarray[10]= tempch[3];
    rxarray[11]= tempch[4];
    rxarray[12]= tempch[5];
    rxarray[13]=',';
    rxarray[14]= 10;
    rxarray[15]= 13;

while (j<16 && i>5)
{
    UART1_Write(rxarray[j]);
    j++;
    flag2=1;
}
if(flag2==1)
{
    PORTD.F2=1;
    flag2=0;
    Delay_ms(100);
}
while (flag3==1)
{
    //UART1_Write('W');
}
if (flag==1)
{
    UART1_Write('$');
    while (cont2<k-1)
    {
        UART1_Write(rxarray[cont2]);
        cont2++;
    }
}

```

```

    }
    UART1_Write(',');
    UART1_Write(10);
    UART1_Write(13);
    cont2=0;
    k=0;
    flag=0;
    flag2=1;
    PORTD.F2=0;
    PORTD.F3=1;
    Delay_ms(500);
    PORTD.F3=0;
}
if(flag2==1)
{
    flag3=1;
    PORTA.F3=1;
    flag2=0;
}
while (flag3==1)
{
//UART1_Write('Q');
}
if (flag==1)
{
    UART1_Write('$');
    while (cont2<k-1)
    {
        UART1_Write(rxarray[cont2]);
        cont2++;
    }
    UART1_Write(',');
    UART1_Write(10);
    UART1_Write(13);
    cont2=0;
    k=0;
    flag=0;
    flag2=0;
    PORTA.F3=0;
}
j=0;
rxarray[0]='$';
rxarray[1]='$';
rxarray[2]='W';
rxarray[3]='I';
rxarray[4]='M';
rxarray[5]='D';
rxarray[6]='U';
rxarray[7]=',';
rxarray[8]='T';
rxarray[9]='O';
rxarray[10]='R';
rxarray[11]='R';
rxarray[12]='E';

```

```

        rxarray[13]=',';
        rxarray[14]='0';
        rxarray[15]='0';
        rxarray[16]='1';
        rxarray[17]=',';
        rxarray[18]= 10;
        rxarray[19]= 13;

        while (j<20 && i>5)
        {
            UART1_Write(rxarray[j]);
            j++;
        }
        j=0;
        i=0;
        cont1=0;
        //UART1_Write('I');
    }
    // UART1_Write('O');
}

}while(1);
}

```

2. Código Fuente Módulo Detección de Tormentas

```

char dato, conta = 0;
char bande1=0, bande2=0;
unsigned char jarray[11];

void main() {
    UART1_Init(19200);
    Delay_ms(100);
    UART2_Init(4800);
    Delay_ms(100);
    UART_Set_Active(&UART1_Read,      &UART1_Write,      &UART1_Data_Ready,
&UART1_Tx_Idle); // set UART1 active

    while(1) {

        if (UART1_Data_Ready()==1 && bande1==0){
            dato = UART1_Read();
            if(dato == 'a') {
                jarray[conta] = dato;
                conta++;
                bande1=1;
            }
        }

        if (UART1_Data_Ready()==1 && bande1==1 && conta<11){
            jarray[conta] = UART1_Read();
            conta++;
        }
    }
}

```

```

    }

    if(conta>=11){ //10
        bande2=1;
        conta=0;
        UART_Set_Active(&UART2_Read, &UART2_Write, &UART2_Data_Ready,
&UART2_Tx_Idle); // set UART2 active
    }

    while(bande2==1 && conta<11){
        UART2_Write(jarray[conta]);
        conta++;
    }

    if(bande2==1 && conta>=11){ //10
        bande1=0;
        bande2=0;
        conta=0;
        UART_Set_Active(&UART1_Read, &UART1_Write, &UART1_Data_Ready,
&UART1_Tx_Idle); // set UART1 active
    }
}
}

```

B. Anexo 2 – Código Fuente Módulo Comunicación y Página Web

1. Código Página Estado Actual

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!--
UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIERÍ
INGENIERÍ MECATRÓNICA
MEGAPROYECTO - ESTACION METEOROLÓGICA - PGINA WEB

JEFFERSON ALEXANDER ESTRADA BARRIOS      05701
NOVIEMBRE DE 2011
JONATHAN ALEJANDRO MANSILLA GAITAN      07095
ABRIL DE 2012
-->
<head>

<?php
include("credenciales.php");
?>

```



```

<title>ESTACIÓN METEOROLÓGICA UVG -</title>

<meta name="viewport" content="width=device-width">
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8" />

<link rel="stylesheet" type="text/css" media="all" href="/css/style.css">
<link rel="stylesheet" type="text/css" media="all" href="/css/personal.css">
  <link rel="stylesheet" id="nivoslider-css" href="/css/nivo-slider.css" type="text/css"
media="all">
  <link rel="stylesheet" id="nivoslider-css" href="/css/default/default.css" type="text/css"
media="all">

<script language="javascript" type="text/javascript" src="/js/jquery.js"></script>
  <script type="text/javascript" src="/fancybox/jquery.mousewheel-3.0.4.pack.js"></script>
  <script type="text/javascript" src="/fancybox/jquery.fancybox-1.3.4.pack.js"></script>
  <link rel="stylesheet" type="text/css" href="/fancybox/jquery.fancybox-1.3.4.css"
media="screen" />
  <script type="text/javascript" src="/js/jquery.nivo.slider.pack.js"></script>
  <script language="javascript" type="text/javascript" src="/js/jquery.flot.js"></script>
  <script language="javascript" type="text/javascript"
src="/js/jquery.flot.selection.js"></script>

```

```
</head>
```

```

<!-- Comienza el código en formato javascript para realizar las peticiones de -->
<!-- Contenido dinamico en la pagina y graficar los datos. -->

```

```

<script>
var vari = "temperatura";
var mensaje = "Temperatura";
var dimensional = " °C";
var maxi;
var data = []; // arreglo con los datos a graficar via AJAX
var overview = "";
var plot = "";
</script>

```

```

<script>
function nextNearest(value, number) {
  var ceil = Math.ceil(value);
  var remainder = value % number;
  if (remainder > 0)
    value = value - remainder + number;
  return value;
}

```

```

}
</script>

```

```

<script type="text/javascript">

function PROMEDIO( array )
{
var largo = array.length;
var total =0;
    for (var i = 0; i < largo; ++i)
        total+=array[i];
return (total/largo).toFixed(2);
}
function MAX_( array ){
    return Math.max.apply( Math, array );
};

function MIN_( array ){
    return Math.min.apply( Math, array );
};
</script>

```

```

<script>

// la funcion doUpdate realiza la peticion de datos a la script "extraerdatos.php"
// para que le devuelva los datos a graficar via AJAX y llama a la funcion Graficar
// que actualiza las graficas con los nuevos datos
function doUpdate(){
function onDataReceived(series)
{
data = series ; // guarda en variable global los datos recibidos
Graficar(data); // actualiza los datos
}
$.ajax({ // funcion ajax que solicita datos al servidor de forma asincrona
url: "aj.php?variable="+vari+"&span="+mensaje,
method: 'GET',
dataType: 'json',
success: onDataReceived
});
};

```

```

function Graficar(series) //metodo que grafica los datos en el area correspondiente
{

```

```

var options = {
lines: { show: true, lineWidth: 2, fill: 1, fillColor: "rgba(233, 237, 244, 0.5)", lineColor: "blue"},
points: { show: false},
xaxis: { ticks: 6,mode: "time", tickLength: 5},
yaxis: { ticks: 8 },
selection: { mode: "x" },
colors: ["#93b2d8"]
};

    var placeholder = $("#placeholder");
    $.plot(placeholder, data, options);
var d1 = [];

                                var tMuestras = series.length;

                                for (var i = 0; i < tMuestras; ++i)
                                d1.push(series[i][1] );
                                //d1 = series[]
                                var lastDato = series[tMuestras-1][1];
                                datos = [series] ;
                                if(vari!="viento_direccion"){
dimensional;                                document.getElementById("maxvalue").innerHTML= MAX_(d1) +
dimensional;                                document.getElementById("minvalue").innerHTML= MIN_(d1) +
                                                }
                                                document.getElementById("promedio").innerHTML=
PROMEDIO(d1) + dimensional;
                                document.getElementById("ultimodato").innerHTML= lastDato +
dimensional;

    $.plot($("#placeholder"), datos, options);
/*
$(function () {
var options = {
lines: { show: true, lineWidth: 2, fill: 1, fillColor: "rgba(233, 237, 244, 0.5)", lineColor: "blue"},
points: { show: false},
xaxis: { ticks: 6,mode: "time", tickLength: 5},
yaxis: { ticks: 8 },
selection: { mode: "x" },
colors: ["#93b2d8"]
};

var placeholder = $("#placeholder");
$.plot(placeholder, [data], options);

```

```

});
*/

}
doUpdate();
window.setInterval(doUpdate, 1000);

/*
$(function ini() {
  var options = {
    //lines: { show: true, fill: false },
    lines: { show: true, lineWidth: 2, fill: 1, fillColor: "rgba(233, 237, 244, 0.9)",
lineColor: "blue"},
    points: { show: false },
    xaxis: { mode: "time", tickLength: 3 },
    selection: { mode: "x" },
    colors: ["#93b2d8", "#93b2d8", "#cb4b4b", "#4da74d", "#9440ed"]

  };
  data = [];
  var placeholder = $("#placeholder");
  $.plot(placeholder, data, options);

  function doUpdate(){
    $.plot(placeholder, data, options);

    function onDataReceived(series) {
      // se llama a la script de php que busca en
      // la base de datos y devuelve los valores almacenados
      // para la variable solicitada

      var d1 = [];
      var tMuestras = series.length;

      for (var i = 0; i < tMuestras; ++i)
        d1.push(series[i][1] );
      //d1 = series[]
      var lastDato = series[tMuestras-1][1];

      data = [series] ;

      if(vari!="viento_direccion"){
        document.getElementById("maxvalue").innerHTML= MAX_(d1) +
dimensional;

        document.getElementById("minvalue").innerHTML= MIN_(d1) +
dimensional;

```

```

    }
    document.getElementById("promedio").innerHTML=
PROMEDIO(d1) + dimensional;
    document.getElementById("ultimodato").innerHTML= lastDato +
dimensional;

```

```

    $.plot($("#placeholder"), data, options);
}
$.ajax({
    url: "aj.php?variable="+vari+"&span="+mensaje,
    method: 'GET',
    dataType: 'json',
    success: onDataReceived
});
};

```

```

doUpdate();
window.setInterval(doUpdate, 10000);
});
*/
</script>

```

```

<script>
// la funcion doUpdate realiza la peticion de datos a la script "extraerdatos.php"
// para que le devuelva los datos a graficar via AJAX y llama a la funcion Graficar
// que actualiza las graficas con los nuevos datos
function ActualizarActual(){

function ActualRecibido(series)
{

var datos = series ; // guarda en variable global los datos recibidos
var data = datos.split(',');

if(data[12]=="Listo"){

document.getElementById("ActualTemp").innerHTML=(data[0]*1.0).toFixed(1)+" °C"; //actualizar
temperatura
//document.getElementById("ActualTemp").innerHTML= nextNearest(data[0], .5) +" °C";
//actualizar temperatura
document.getElementById("ActualVientoD").innerHTML=data[1]; //actualizar direccion del viento

```

```

document.getElementById("ActualVientoV").innerHTML=(data[2]*3.6).toFixed(2) + " Km/h";
//actualizar velocidad viento
document.getElementById("ActualHumedad").innerHTML=data[3] + "%"; //actualizar humedad
document.getElementById("ActualPresion").innerHTML=data[4] + " Bar"; //actualizar presion
document.getElementById("imagen").innerHTML='';
document.getElementById("ActualRocio").innerHTML=data[5] + " °C"; //actualizar punto de rocio
document.getElementById("ActualRadiacion").innerHTML=(data[6]*1.0).toFixed(2) + " watt/m^2";
//actualizar radiacion
document.getElementById("sismo").innerHTML=(data[13]); //actualizar radiacion
document.getElementById("campoE").innerHTML=data[14] + " kV/m"; //actualizar campo
electrostatico

}

if(data[7]!=""){
document.getElementById("fechas").innerHTML= "Datos Recopilados hoy "+data[7] + ", hora:
"+data[8]; //actualizar radiacion
else{document.getElementById("fechas").innerHTML= "No hay datos recopilados Hoy";}

}

$.ajax({ // funcion ajax que solicita datos al servidor de forma asincrona
url: "extraerActual.php?Cache="+parseInt(Math.random() * 99999999),
method: 'GET',
success: ActualRecibido
});

};

ActualizarActual(); // actualizar las variables mas recientes
window.setInterval(ActualizarActual, 10000);
</script>

```

```

<script>
function viewTemperatura()
{
if(vari!="temperatura")
{
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "temperatura";
mensaje = "Temperatura";
dimensional = " °C";
document.getElementById("maxspan").innerHTML="Temperatura Máxima";
document.getElementById("maxvalue").innerHTML= "N/A";

```

```

document.getElementById("minspan").innerHTML="Temperatura Mnima";
document.getElementById("minvalue").innerHTML= "N/A";
document.getElementById("charttitle").innerHTML="Temperaturas mostradas en C";
document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
});
}

```

```

function viewHumedad()
{
if(vari!="humedad_relativa"){
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "humedad_relativa";
mensaje = "Humedad Relativa";
dimensional = " %";
document.getElementById("maxspan").innerHTML="Humedad Mxima";
document.getElementById("maxvalue").innerHTML= "N/A";
document.getElementById("minspan").innerHTML="Humedad Mnima";
document.getElementById("minvalue").innerHTML= "N/A";
document.getElementById("charttitle").innerHTML="Humedad Mostrada en Porcentaje %";
document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
});
}

```

```

function viewVientoV()
{
if(vari!="viento_velocidad"){
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "viento_velocidad";
mensaje = "Velocidad del Viento";
dimensional = " m/s";
document.getElementById("maxspan").innerHTML="Velocidad Mxima";
document.getElementById("maxvalue").innerHTML= "N/A";
document.getElementById("minspan").innerHTML="Velocidad Mnima";
document.getElementById("minvalue").innerHTML= "N/A";
document.getElementById("charttitle").innerHTML="Velocidad Mostrada en metros por segundo
m/s";

```

```

document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
});
}

```

```

function viewVientoD()
{
if(vari!="viento_direccion"){
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "viento_direccion";
mensaje = "Dirección del Viento";
dimensional = " grados";
document.getElementById("maxspan").innerHTML=" ";
document.getElementById("maxvalue").innerHTML= " ";
document.getElementById("minspan").innerHTML=" ";
document.getElementById("minvalue").innerHTML= " ";
document.getElementById("chartitle").innerHTML="Dirección Mostrada en grados ";
document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
});
}

```

```

function viewRadiacion()
{
if(vari!="radiacion"){
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "radiacion";
mensaje = "Radiación";
dimensional = " watt/m^2";
document.getElementById("maxspan").innerHTML="Radiación Máxima ";
document.getElementById("maxvalue").innerHTML= "N/A";
document.getElementById("minspan").innerHTML="Radiación Mínima ";
document.getElementById("minvalue").innerHTML= "N/A ";
document.getElementById("chartitle").innerHTML="Radiación Mostrada en watt/m^2 ";
document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}

```



```

}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
});
}

```

```

function viewPresion()
{
    if(vari!="presion"){
        document.getElementById("placeholder").innerHTML="<div align='center'> <img
        src='./images/loading.gif'> </div>";
        vari = "presion";
        mensaje = "Presión";
        dimensional = " bar";
        document.getElementById("maxspan").innerHTML="Presión Máxima ";
        document.getElementById("maxvalue").innerHTML= "N/A";
        document.getElementById("minspan").innerHTML="Presión Mínima ";
        document.getElementById("minvalue").innerHTML= "N/A ";
        document.getElementById("chartitle").innerHTML="Presión Mostrada en bar ";
        document.getElementById("ultimodato").innerHTML= "N/A";
        document.getElementById("promedio").innerHTML= "N/A";
    }
    doUpdate();
    jQuery(document).ready(function() {
        $("#launch").trigger('click');
    });
}

```

```

function viewRocio()
{
    if(vari!="punto_rocio"){
        document.getElementById("placeholder").innerHTML="<div align='center'> <img
        src='./images/loading.gif'> </div>";
        vari = "punto_rocio";
        mensaje = "Punto de Rocío";
        dimensional = " °C";
        document.getElementById("maxspan").innerHTML="Punto de Rocío Máxima ";
        document.getElementById("maxvalue").innerHTML= "N/A";
        document.getElementById("minspan").innerHTML="Punto de Rocío Mínima ";
        document.getElementById("minvalue").innerHTML= "N/A ";
        document.getElementById("chartitle").innerHTML="Punto de Rocío Mostrada en °C ";
        document.getElementById("ultimodato").innerHTML= "N/A";
        document.getElementById("promedio").innerHTML= "N/A";
    }
    doUpdate();
    jQuery(document).ready(function() {

```

```

    $("#launch").trigger('click');
  });
}

```

```

function viewCampoE()
{
if(vari!="campo_electros"){
document.getElementById("placeholder").innerHTML="<div align='center'> <img
src='./images/loading.gif'> </div>";
vari = "campo_electros";
mensaje = "Campo Electrostático";
dimensional = " kV/m";
document.getElementById("maxspan").innerHTML="Campo Electrostático Máximo ";
document.getElementById("maxvalue").innerHTML= "N/A";
document.getElementById("minspan").innerHTML="Campo Electrostático Mínimo ";
document.getElementById("minvalue").innerHTML= "N/A ";
document.getElementById("chartitle").innerHTML="Campo Electrostático Mostrado en kV/m ";
document.getElementById("ultimodato").innerHTML= "N/A";
document.getElementById("promedio").innerHTML= "N/A";
}
doUpdate();
jQuery(document).ready(function() {
    $("#launch").trigger('click');
  });
}

```

```

</script>

```

```

<script type="text/javascript">
    $(document).ready(function() {

        $("a.get_info").fancybox({
            'overlayShow' : true,
            'transitionIn' : 'elastic',
            'transitionOut' : 'elastic',
            'titlePosition' : 'inside',
            'overlayColor' : '#000',
            'overlayOpacity' : 0.7,
            'autoScale': true
        });
    });
</script>

```

```

<script type="text/javascript">
jQuery(window).load(function(){
    jQuery("#nivoslider-54").nivoSlider({

```

```

    effect:"random",
    slices:15,
    boxCols:8,
    boxRows:4,
    animSpeed:1000,
    pauseTime:7000,
    startSlide:0,
    directionNav:true,
    directionNavHide:true,
    controlNav:false,
    controlNavThumbs:false,
    controlNavThumbsFromRel:true,
    keyboardNav:true,
    pauseOnHover:true,
    manualAdvance:false
  });
});
</script>

```

```
<body >
```

```

    <div style="width:960px; margin:auto; margin-top:0px;">
    
    </div>

    <div style="width:960px; height:215px; position:relative; top:2px; margin:auto; ">
      <div class="slider-wrapper theme-default">
        <div class="ribbon"></div>
        <div id="nivoslider-54" class="nivoSlider">
          
          
          <a href="http://uvg.edu.gt"></a>
          
          
        </div>
        <div id="htmlcaption" class="nivo-html-caption">
          Visitanos <a href="http://www.uvg.edu.gt">
www.uvg.edu.gt</a>.
        </div>

```

```

        </div>
    </div>

<div id="wrapper">

<header id="header"> <!-- Se muestra la barra de links, con el acceso a todas las secciones de la
pagina-->
<nav id="nav">
<div class="menu-main-menu-container">
<ul id="menu-main-menu" class="menu">
<li class="menu-item"><a href="/nosotros.php"> ¿Quiénes Somos?</a></li>
<li class="menu-item current-menu-item current_page_item"><a href="/index.php"> Estado
Actual</a></li>
<li class="menu-item "><a href="/historial.php"> Historial</a></li>
<li class="menu-item"><a href="/acercade.php"> Acerca del Proyecto</a></li>
<li class="menu-item"><a href="/imagenes.php"> Imágenes</a></li>
</ul></div></nav>
</header>

<div id="main">
    <div id="primary">

        <div id="subheading">
            <h2> Estado Actual de las Variables Climáticas </h2>
        </div>

        <div id="update" align="center" style="margin-top:20px;">

            <input id="btnTemp" type="button" onclick="viewTemperatura()" value="Temperatura">
            <input id="btnHumedad" type="button" onclick="viewHumedad()" value="Humedad
Relativa">
            <input id="btnHumedad" type="button" onclick="viewVientoV()"
value="Velocidad Viento">
            <input id="btnHumedad" type="button" onclick="viewVientoD()"
value="Dirección Viento">
            <input id="btnHumedad" type="button" onclick="viewRadiacion()"
value="Radiación">
            <input id="btnHumedad" type="button" onclick="viewPresion()"
value="Presión">
            <input id="btnHumedad" type="button" onclick="viewRocio()"
value="Punto de Rocío">
            <input id="btnHumedad" type="button" onclick="viewCampoE()"
value="Campo Electrostático">

```

```

</div>

<div align="left" style="float:left; margin-top:25px; padding-
left:55px; min-width:60px;">
  <a class="get_info" id="launch" href="#GRTemp" ">
  </a>
</div>

<div class="variablesT" style=" margin-top:20px;">

  <table align="center" border="0">
  <tr>

    <td>
      <div id="imagen"
style="width:250px;"> </div>

    </td>

    <td >

      <h3 id="ActualTemp" class="temperatura"
title="Temperatura -Click para ver el comportamiento de hoy" onclick="viewTemperatura()"> -- °C
</h3>

      <h3>Viento: <span id="ActualVientoD"
class="variables"> -- </span> <span id="ActualVientoV" class="variables"> </span> </h3>
      <h3>Humedad: <span
id="ActualHumedad" class="variables"> -- </span> <span id="ActualVientoV" > </span>
</h3>

    </td>

  </tr>
</table>

</div>

<div id="OtrasVariables" style="margin-
top:25px;">

  <table align="center">
  <tr>

```

```

<!-- Mostrar la variable Presion -->
<td >
<div class="variables1" style=" margin-top:20px;">

    <h4> Presión: </h4>

    <span id="ActualPresion"
class="variables"> N/A </span>

</div>
</td>
<!-- Mostrar la variable Radiacion -->
<td>
<div class="variables1" style=" margin-top:20px;">

    <h4> Radiación: </h4>
    <span id="ActualRadiacion"
class="variables"> N/A </span>

</div>
</td>
</tr><tr>
<!-- Mostrar la variable Punto de Rocío -->
<td>
<div class="variables1" style="margin-top:20px;">

    <h4 > Punto de Rocío: </h4>
    <span id="ActualRocio" class="variables">
N/A </span>

</div>
</td>

<!-- Mostrar la variable Campo electro Estatico -->
<td>
<div class="variables1" style="margin-top:20px;">

    <h4 > Campo Electrostático: </h4>

```

```

</span>
<span id="campoE" class="variables"> N/A
</div>
</td>
</tr>
</table>
</div>

```

```

<div style="margin-left:150px; margin-top:25px;">
  <h3>Información de Sismos:</h3>
  <ul style="font-size:16px">
    Última intensidad registrada en escala de Mercalli
  </ul>
  Modificada: <span id="sismo" style="position: relative; "> N/A</span>
</div>

```

```

<div style="position:relative; margin-top:60px;">
  <h3>¿Qué es esto?</h3>
  <ul>
    <li>Son los datos más recientes recopilados por la estación meteorológica UVG. <span
      class="lastUpdate" style="position: relative; "> </span></li>
  </ul>

```

```

<h3 id="fechas" >Datos tomados hoy 11/17/2011, hora: 12:38 AM</h3>
</div>
</div>
</div>

```

```

        <div id="maindivHistorico" class="historicos" style="width:650px;
display:none;" >

        <div id="GRTemp" align="center" style=";" align="center"
style="psition:absolute">

        <h2 class="cblue" id="chartitle"> Temperaturas mostradas en °C </h2>
<div id="placeholder" style="width:600px;height:320px;"></div>

<div class="maxmin" style=" width:550px; margin-top:20px;">

        <table>
        <tr>

        <td style="padding:10px;">
        <p>
<span class="cblue" id="maxspan"> Temperatura maxima </span>
<span class="cblack" id="maxvalue"> el valor </span>
        </p>

        <p>
        <span class="cblue" id="minspan"> Temperatura minima </span>
<span class="cblack" id="minvalue"> el valor </span>
        </p>
        </td>

        <td style="padding:10px;">
        <p>
<span class="cblue" > Promedio </span>
<span class="cblack" id="promedio"> el valor </span>
        </p>

        <p>
        <span class="cblue" > Último dato </span>
<span class="cblack" id="ultimodato"> el valor </span>
        </p>
        </td>

        </tr>

        </table>

```



```
</div>
```

```
</div>
```

```
</div>
```

```
<footer id="footer">
<p>Torre Meteorológica UVG</p>
<!--
      <ul id="menu-footer-menu" class="menu"><li id="menu-item-340" class="menu-
item-first menu-item menu-item-type-post_type menu-item-object-page menu-item-340"><a
href="http://nivo.dev7studios.com/license/">License</a> |</li><li id="menu-item-343"
class="menu-item menu-item-type-post_type menu-item-object-page menu-item-343"><a
href="http://nivo.dev7studios.com/refunds/">Refunds</a> |</li><li id="menu-item-342"
class="menu-item menu-item-type-post_type menu-item-object-page menu-item-342"><a
href="http://nivo.dev7studios.com/privacy/">Privacy</a> |</li><li id="menu-item-344"
class="menu-item-last menu-item menu-item-type-post_type menu-item-object-page menu-item-
344"><a href="http://nivo.dev7studios.com/contact/">Contact</a> |</li></ul>
-->
      <p class="bottom">Departamento de Ingeniería Mecatrónica</p>
</footer>
```

```
</body>
```

```
</html>
```

2. Código Página Historial

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!--
UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIER´
INGENIER´ MECATR´MICA
MEGAPROYECTO - ESTACI´M METEOROLOGICA - PGINA WEB

JEFFERSON ALEXANDER ESTRADA BARRIOS          05701
OCTUBRE DE 2011
JONATHAN ALEJANDRO MANSILLA GAITAN          07095
ABRIL DE 2012
-->
```

```

<html>
<head>
<?php
// se cargan las credenciales con la informacion de
include("credenciales.php");
?>
<meta name="viewport" content="width=device-width">
<title>ESTACIÓN METEOROLÓGICA UVG -HISTORIAL-</title>

<!-- Se importan los archivos de estilos en cascada CSS para el formato -->
<style type="text/css" title="currentStyle">
@import "./css/demo_table_jui.css";
@import "./css/TableTools.css";
@import "./css/demos.css";
@import "./css/themes/blitzer/jquery-ui-1.8.13.custom.css";
@import "./css/style.css";
@import "./css/personal.css";
@import "./css/nivo-slider.css";
@import "./css/default/default.css";
</style>

<!-- Se importan todas las librerias JAVASCRIPT y JQUERY que se usan en el proyecto -->
<script type="text/javascript" language="javascript" src="./media/js/jquery.js"></script>
<script src="./js/jquery-ui.min.js"></script>

<script type="text/javascript" language="javascript"
src="./media/js/jquery.dataTables.js"></script>
<script type="text/javascript" charset="utf-8" src="./media/js/ZeroClipboard.js"></script>
<script type="text/javascript" charset="utf-8" src="./media/js/TableTools.js"></script>
<script type="text/javascript" src="./js/jquery.nivo.slider.pack.js"></script>

<script type="text/javascript" src="./js/jquery.flot.js"></script>
<script type="text/javascript" src="./js/jquery.flot.selection.js"></script>
<script type="text/javascript" src="./js/ui/jquery.ui.core.js"></script>
<script type="text/javascript" src="./js/ui/jquery.ui.widget.js"></script>
<script type="text/javascript" src="./js/ui/jquery.ui.datepicker.js"></script>
<script type="text/javascript" src="./js/ui/jquery.ui.button.js"></script>

<!-- Comienza el código en formato javascript para realizar las peticiones de -->
<!-- Contenido dinamico en la pagina y graficar los datos. -->
<script>
// habilitar la funcion del calendario emergente en el
// textbox para las fechas iniciales y finales de busqueda
$(function() {
$( "#datepicker" ).datepicker();

```

```
$( "#datepicker2" ).datepicker();
});
```

```
// habilitar los radio button para utilizar la libreria de
// jquery y mejorar el aspecto predeterminado
$(function() {
$( "#radio" ).buttonset();
});
```

```
<?php date_default_timezone_set("America/Guatemala"); ?>
// variables globales que se usan en la pagina
var vari = "temperatura"; // para indicar la variable a graficar
var date1 = <?php $tomorrow = mktime(0,0,0,date("m"),date("d")-7,date("Y")); echo "".date("Y-
m-d", $tomorrow).";";?> // fecha inicial de las busquedas
var date2 = <?php echo "".date("Y-m-d").";";?> // fecha final de las busquedas
var data = []; // arreglo con los datos a graficar via AJAX
var overview = "";
var plot = "";
```

```
function rTemp() // graficar los datos de temperatura
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "temperatura";
document.getElementById("mainTitle").innerHTML = "Temperatura ( °C )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}
```

```
function rPresion() // graficar los datos de presion
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "presion";
document.getElementById("mainTitle").innerHTML = "Presión ( bar )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}
```

```
function rVientoDir() // graficar la direccion del viento
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "viento_direccion";
document.getElementById("mainTitle").innerHTML = "Direccion del Viento ( grados )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
```

```

}

function rVientoVel() // graficar la velocidad del viento
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "viento_velocidad";
document.getElementById("mainTitle").innerHTML = "Velocidad del Viento ( m/s )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}

function rRadiacion() // graficar la radiacion
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "radiacion";
document.getElementById("mainTitle").innerHTML = "Radiación ( Watt/m^2 )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}

function rHumedad() // graficar la humedad relativa
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "humedad_relativa";
document.getElementById("mainTitle").innerHTML = "Humedad Relativa ( % )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}

function rPuntoRocio() // graficar el punto de rocío calculado
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "punto_rocio";
document.getElementById("mainTitle").innerHTML = "Punto de Rocío ( °C )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}

function rCamposE() // graficar el campo electrostatico medido
{
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
vari = "campo_electros";
}

```

```
document.getElementById("mainTitle").innerHTML = "Campo Electrostático ( kV/m )";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
}
```

```
// la funcion doUpdate realiza la peticion de datos a la script "extraerdatos.php"
// para que le devuelva los datos a graficar via AJAX y llama a la funcion Graficar
// que actualiza las graficas con los nuevos datos
```

```
function doUpdate(){
function onDataReceived(series)
{
data = [series] ; // guarda en variable global los datos recibidos
Graficar(); // actualiza los datos
}
$.ajax({ // funcion ajax que solicita datos al servidor de forma asincrona
url: "extraerdatos.php?variable="+vari+"&date1="+date1+"&date2="+date2,
method: 'GET',
dataType: 'json',
success: onDataReceived
});
};
```

```
// la funcion doUpdate2 se encarga de pedir los datos para actualizar la tabla, la diferencia
// con doUpdate es que se devuelve una tabla completa, y se actualiza el contenido en la
// seccion disponible para la tabla. tambien se actualiza la referencia JQUERY para que se
// pueda navegar en la tabla y realizar la descarga de los datos.
```

```
function doUpdate2(){
function onDataReceived(series) {
```

```
document.getElementById("tablaDatos").innerHTML = series;
```

```
$('.display').dataTable
( {
"bjQueryUI": true,
"sPaginationType": "full_numbers",
"sDom": '<H"TI>t<F"ip>',
"oTableTools":
{
"aButtons": [{
"sExtends": "collection",
"sButtonText": " Descargar Datos",
"aButtons": [ "xls", "pdf" ]
}
}
});
```

```
}
$.ajax({ // funcion ajax que solicita datos al servidor de forma asincrona
url: "extraerTabla.php?variable="+vari+"&date1="+date1+"&date2="+date2,
method: 'GET',
```

```

success: onDataReceived
});
};

```

```

function Graficar() //metodo que grafica los datos en el area correspondiente
{
$("#area_graficas").fadeOut(10);
$(function () {
var options = {
lines: { show: true, lineWidth: 2, fill: 1, fillColor: "rgba(233, 237, 244, 0.5)", lineColor: "blue"},
points: { show: false},
xaxis: { ticks: 6, mode: "time", tickLength: 5},
yaxis: { ticks: 8 },
selection: { mode: "x" },
colors: ["#93b2d8"]
};
plot = $.plot($("#placeholder"), data, options);
overview = $.plot($("#overview"), data, {
series: {
lines: { show: true, lineWidth: 1 },
shadowSize: 0
},
xaxis: { ticks: [], mode: "time" },
yaxis: { ticks: [], min: 0, autoscaleMargin: 0.1 },
selection: { mode: "x" }
});

```

```

// placeholder es la division en la que se grafican los datos
$("#placeholder").bind("plotselected", function (event, ranges) {
// hacer el zoom en el area seleccionada
plot = $.plot($("#placeholder"), data,
$.extend(true, {}, options, {
xaxis: { min: ranges.xaxis.from, max: ranges.xaxis.to }
}));
overview.setSelection(ranges, true);
});
// overview es la grafica pequeña que sirve para visualizar
// todos los datos de una mejor manera
$("#overview").bind("plotselected", function (event, ranges) {
plot.setSelection(ranges);
});
//$("#area_graficas").show("scale", {}, 1000);
$("#area_graficas").fadeIn(1000);
}); }

```

```

doUpdate(); // actualizar la grafica al cargar la pagina
doUpdate2(); // actualizar la tabla al cargar la pagina

```

```

// ObtenerDatos es una funcion que realiza la actualizacion de la
// grafica y la tabla dependiendo de las fechas seleccionadas por
// el usuario.
function ObtenerDatos(boton)
{
date1=document.getElementById("datepicker").value+"";
date2=document.getElementById("datepicker2").value+"";

var d1 = new Date(date1); // crea un objeto date para comparara las fechas
var d2 = new Date(date2); // crea un objeto date para comparara las fechas

var milli_d1 = d1.getTime();
var milli_d2 = d2.getTime();

if(milli_d1>milli_d2)
{
alert("La fecha inicial debe ser menor a la final");
return false;
}
var date_array =date1.split("/"); // separar la fecha para cambiar el formato
date1 = (date_array[2] + "-" +date_array[0] + "-" + date_array[1]);
var date_array =date2.split("/"); // separar la fecha para cambiar el formato
date2 = (date_array[2] + "-" +date_array[0] + "-" + date_array[1]);
alert("Mostrar datos desde fecha: "+date1+" hasta "+date2);
document.getElementById("placeholder").innerHTML = "<div align='center'> <img
src='./images/loading.gif'> </div>";
document.getElementById("overview").innerHTML = "";
setTimeout("doUpdate()", 200);//actualiza la grafica luego de la animacion
setTimeout("doUpdate2()", 200);//actualiza la tabla luego de la animacion
}
</script>

```

<!-- Para mostrar la animacion de publicidad se utiliza la libreria de NIVO-SLIDER, no se utiliza flash por lo que es compatible con todos los dispositivos que soporten un navegador web -->

```

<script type="text/javascript">
jQuery(window).load(function(){
jQuery("#nivoslider-54").nivoSlider({
effect:"random",
slices:15,
boxCols:8,
boxRows:4,
animSpeed:500,
pauseTime:3000,
startSlide:0,
directionNav:true,
directionNavHide:true,

```

```

controlNav:false,
controlNavThumbs:false,
controlNavThumbsFromRel:true,
keyboardNav:true,
pauseOnHover:true,
manualAdvance:false
});
});
</script>
</head>

<body>

<div style="width:960px; margin:auto; margin-top:0px;">
    
</div>

    <div style="width:960px; height:215px; position:relative; top:2px; margin:auto; ">
        <div class="slider-wrapper theme-default">
            <div class="ribbon"></div>
            <div id="nivoslider-54" class="nivoSlider">
                
                
                <a href="http://uvg.edu.gt"></a>
                
                
            </div>
            <div id="htmlcaption" class="nivo-html-caption">
                Visitanos <a href="http://www.uvg.edu.gt">
www.uvg.edu.gt</a>.
            </div>
        </div>
    </div>

<div id="wrapper">

<header id="header"> <!-- Se muestra la barra de links, con el acceso a todas las secciones de la
pagina-->

<nav id="nav">
<div class="menu-main-menu-container">
<ul id="menu-main-menu" class="menu">
<li class="menu-item"><a href="./nosotros.php"> ¿Quiénes Somos?</a></li>
<li class="menu-item "><a href="./index.php"> Estado Actual</a></li>

```



```

<li class="menu-item current-menu-item current_page_item"><a href="/historial.php">
Historial</a></li>
<li class="menu-item"><a href="/acercade.php"> Acerca del Proyecto</a></li>
<li class="menu-item"><a href="/imagenes.php"> Imágenes</a></li>
</ul></div></nav>
</header>
</div>

```

```

<div id="main"> <!-- Division principal, contiene todos los componentes de la seccion historial de
la pagina -->

```

```

<div id="subheading">
<h2>INFORMACIÓN RECOPIADA </h2>
</div>

```

```

<!--
*****
***** -->

```

```

<!-- Conjunto de radio button para elegir la variable que se quiere graficar, utilizan la libreria -->
<!-- radio de JQUERY para mejorar el aspecto por defecto y dar una apariencia mas profesional
-->

```

```

<div align="center" style="margin-top:10px;">
<div id="radio">
<input type="radio" onClick="rTemp()" id="radio1" name="radio" checked="checked" /><label
for="radio1">Temperatura</label>
<input type="radio" onClick="rPresion()" id="radio2" name="radio" /><label
for="radio2">Presión</label>
<input type="radio" onClick="rVientoVel()" id="radio3" name="radio" /><label
for="radio3">Velocidad Viento</label>
<input type="radio" onClick="rVientoDir()" id="radio4" name="radio" /><label
for="radio4">Dirección del Viento</label>
<input type="radio" onClick="rHumedad()" id="radio5" name="radio" /><label
for="radio5">Humedad Relativa</label>
<input type="radio" onClick="rPuntoRocio()" id="radio6" name="radio" /><label for="radio6">P.
Rocío</label>
<input type="radio" onClick="rRadiacion()" id="radio7" name="radio" /><label
for="radio7">Radiación</label>
<input type="radio" onClick="rCamposE()" id="radio8" name="radio" /><label
for="radio8">Campo Electrostático</label>
</div>
</div>

```

```

<div align="center" style="margin-top:20px;" >
<div align="center" id="area_graficas" style="height:500px;" >

```

```

<h1 class="cblue" id="mainTitle"> Temperatura °C </h1>
<div id="placeholder" style="width:600px;height:300px;"></div>
<h1 class="cblue" >Navegar por toda la información recopilada</h1>
<div id="overview" style="margin-top:20px;width:600px;height:60px;"></div>
</div>

<!-- Mostrar el formulario con los campos para solicitar datos en -->
<!-- una fecha determinada -->
<h1 class="cblue"> Buscar datos por fecha </h1>
<div align="center" style="border:1px solid silver; padding:20px; width:600px;">
<table style="position: relative ; " >
<tr>
<td width ="20%" style="vertical-align: middle;">
<!-- mostrar fecha inicial titulo-->
<span class="fechas" >
<b>Fecha Inicial</b> </span>
</td>

<td width="20%" style="vertical-align: middle;">
<!-- mostrar dialogo para elegir la fecha-->
<input type="text" id="datepicker" size="15" >
</td>

<td width="20%" style="vertical-align: middle;">
<!-- mostrar dialogo para elegir la fecha inicial-->
<span class="fechas" > <b>Fecha Final</b> </span>
</td>

<td width="20%" style="vertical-align: middle;">
<!-- mostrar dialogo para elegir la fecha final-->
<input type="text" id="datepicker2" size="15" >

<script>
// ingresar a los campos de busqueda la fecha actual
var date_array =date1.split("-"); // separar la fecha para cambiar el formato
document.getElementById("datepicker").value = (date_array[1] + "/" +date_array[2] + "/" +
date_array[0]);
var date_array =date2.split("-"); // separa la fecha para cambiar el formato
document.getElementById("datepicker2").value = (date_array[1] + "/" +date_array[2] + "/" +
date_array[0]);
</script>

</td>

<!-- Boton de Buscar Datos por fecha, llama a la funcion de ObtenerDatos -->
<td width="20%" style="vertical-align: middle;">
<input id="Procesar" type="button" onclick="ObtenerDatos()" value="BUSCAR"
style="padding:15px;">

```

```
</td>
```

```
</font></td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
<!-- AREA DESTINADA A MOSTRAR LA TABLA DE DATOS -->
```

```
<!-- Se encuentra vacia, el contenido se actualiza dinamicamente -->
```

```
<!-- utilizando AJAX llamado desde la funcion doUpdate2() -->
```

```
<div id="tablaDatos" style="margin-top:20px;" ></div>
```

```
</div>
```

```
<!-- mostrar el pie de pagina, con la informacion acerca del proyecto -->
```

```
<footer id="footer">
```

```
<p > <b> Torre Meteorológica UVG </b>
```

```
<p >Departamento de Ingeniería Mecatrónica</p>
```

```
<p >Desarrollado por Jefferson Estrada</p>
```

```
</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

XII. GLOSARIO

Baud Rate: La tasa de baudios es el número de unidades de señal por segundo. Un baudio puede contener varios bits.

Capacitancia: es la propiedad que tienen los cuerpos para mantener una carga eléctrica.

CMOS: o Complementary metal-oxide-semiconductor por sus siglas en inglés, es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS configurados de tal forma que, en estado de reposo, el consumo de energía es únicamente el debido a las corrientes parásitas. En la actualidad, la mayoría de los circuitos integrados que se fabrican utilizan la tecnología CMOS, por ejemplo, microprocesadores, memorias, y muchos otros circuitos integrados digitales cuyo consumo es considerablemente bajo.

Corriente Eléctrica: es el flujo de carga por unidad de tiempo que recorre un material. Se debe al movimiento de los electrones en el interior del material.

Empaquetado DIP: o Dual in-line package por sus siglas en inglés, es una forma de encapsulamiento común en la construcción de circuitos integrados que consiste en un bloque con dos hileras paralelas de pines, la cantidad de éstos depende de cada circuito. Por la posición y espaciamiento entre pines, son prácticos para construir prototipos en tablillas de protoboard.

GSM: El sistema global para las comunicaciones móviles o Global System for Mobile Communications por sus siglas en inglés, es un sistema estándar, libre de regalías, de telefonía móvil digital. Se considera, por su velocidad de transmisión y otras características, un estándar de segunda generación (2G).

Handshaking: Proceso de negociación que dinámicamente establece los parámetros o “reglas” del proceso de comunicación entre dos dispositivos antes de que estos comiencen a comunicarse normalmente entre ellos.

Impedancia: es la oposición al paso de la corriente alterna. A diferencia de la resistencia, la impedancia incluye los efectos de acumulación y eliminación de carga (capacitancia) e/o inducción magnética (inductancia). Este efecto es apreciable al analizar la señal eléctrica implicada en el tiempo.

Inductancia: es una medida de la oposición a un cambio de corriente de un inductor o bobina que almacena energía en presencia de un campo magnético, y se define como la relación entre el flujo magnético (Φ) y la intensidad de corriente eléctrica (I) que circula por la bobina y el número de vueltas (N) de el devanado:

$$L = \frac{\Phi N}{I}$$

La inductancia depende de las características físicas del conductor y de la longitud del mismo.

Memoria EEPROM: Electrically Erasable Programmable Read-Only Memory, por sus siglas en inglés. Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente.

Memoria Flash: permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Derivada de la memoria EEPROM.

Memoria RAM: La memoria de acceso aleatorio (random-access memory por sus siglas en inglés) se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan “de acceso aleatorio” porque se puede leer o escribir en una posición

de memoria no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible.

Memoria ROM: memoria de solo lectura o read-only memory por sus siglas en inglés. Es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite sólo la lectura de la información y no su escritura, independientemente de la presencia o no de una fuente de energía.

MÓDEM: o Modulador Demodulador, es un dispositivo que sirve para enviar una señal llamada moduladora mediante otra señal llamada portadora. Su uso más común y conocido es la transmisión de datos, generalmente por vía telefónica.

TCP/IP: La familia de protocolos de Internet es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, a los dos protocolos más importantes que la componen: *Protocolo de Control de Transmisión* (TCP) y *Protocolo de Internet* (IP), que fueron dos de los primeros en definirse, y que son los más utilizados de la familia.

Transistor: es un dispositivo electrónico utilizado para producir una señal de salida en respuesta a otra señal de entrada. Cumple funciones de amplificador, oscilador, conmutador o rectificador. Actualmente se encuentran prácticamente en todos los aparatos electrónicos de uso diario: radios, televisores, reproductores de audio y video, relojes de cuarzo, computadoras, lámparas fluorescentes, tomógrafos, teléfonos celulares, etc.

TTL: Lógica Transistor a Transistor o Transistor-Transistor Logic por sus siglas en inglés, es una tecnología de construcción de circuitos electrónicos digitales. En los componentes fabricados con tecnología TTL los elementos de entrada y salida del dispositivo son transistores bipolares.

UART: son las siglas de "Universal Asynchronous Receiver-Transmitter". Éste controla los puertos y dispositivos seriales. Las funciones principales de UART son de manejar las interrupciones de los dispositivos conectados al puerto serial y de convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serial, para que puedan ser transmitidos a través de los puertos y viceversa.

USART: es el acrónimo de Universal Synchronous/Asynchronous Receiver Transmitter. Se trata de un periférico para la transmisión de datos en formato serial, utilizando técnicas de transmisión sincrónica o asincrónica, según se configure dicho periférico.

V AC: Voltaje de Corriente Alterna.

V DC: Voltaje de Corriente Continua.

Web: Procede de "World Wide Web" se refiere a una red que comprende el conjunto de todas las páginas que pueden consultarse a través de internet.

XIII. FICHA TÉCNICA DEL SISTEMA



Lightning Prevention

El Sistema Previstorm es el único capaz de detectar de forma preventiva el fenómeno del rayo. No es necesaria la caída de un rayo para que detecte algún tipo de actividad tormentosa.⁽⁹⁾

Tabla 10. Especificaciones técnicas⁽¹⁵⁾

Alimentación	125/220 V AC a 24 V DC o alimentación ininterrumpida a 24 V DC
Relés	4 Relés (3 Alarmas y 1 Error)
Tiempo Fin de Alarma	Configurable
Radio de Detección	30 Kms
Fijación Módulo	Mural o Sobremesa
Señal Activación Alarma	2 "pitidos"
Señal Desactivación Alarma	1 "pitidos"
Comunicación PC	Puerto RS-232

Características

Está formado por:

- Un **sensor** exterior **MCE** (Medidor de Campo Electrostatico), tipo molino de campo, que permite medir de forma dinámica la evolución del campo electrostático, mediante el diferencial de potencial existente entre la nube y el potencial cero de la tierra por unidad de longitud.⁽¹⁵⁾
- Un **Módulo adquisición de datos (MAD)** que permite:
 - El Control y Configuración del Sistema.
 - La Visualización del Valor de Campo.
 - Disparo de los Relés asociados a cada Nivel de Alarma.
 - La Visualización de los Parámetros de Configuración.⁽¹⁵⁾

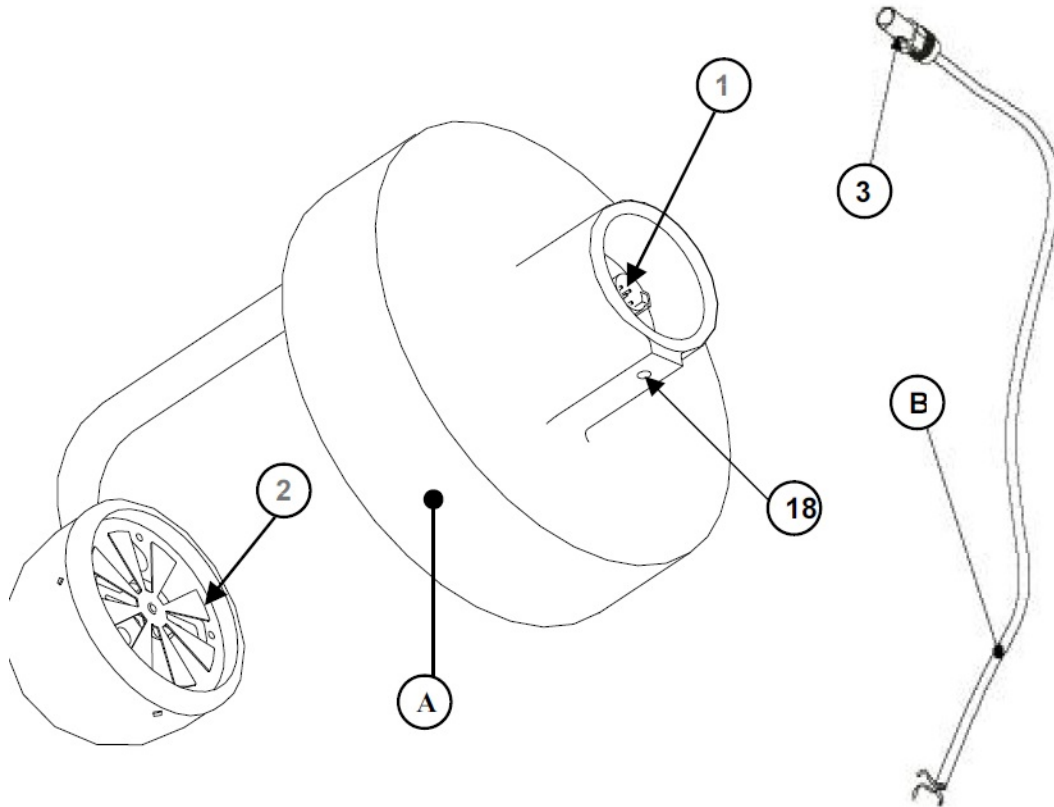
Ventajas

Es un valioso sistema que permite disponer de información en tiempo real de la evolución de la tormenta y caída de rayos, durante 24 horas al día, nos permite adelantarnos con total seguridad a la primera descarga electro atmosférica y de forma preventiva y en general:

- a) **Activar** los protocolos de seguridad.
- b) **Asegurar** la protección de las personas.
- c) **Garantizar** la continuidad en el servicio y evitar pérdidas en producción.
- d) **Ampliar** el nivel de protección.

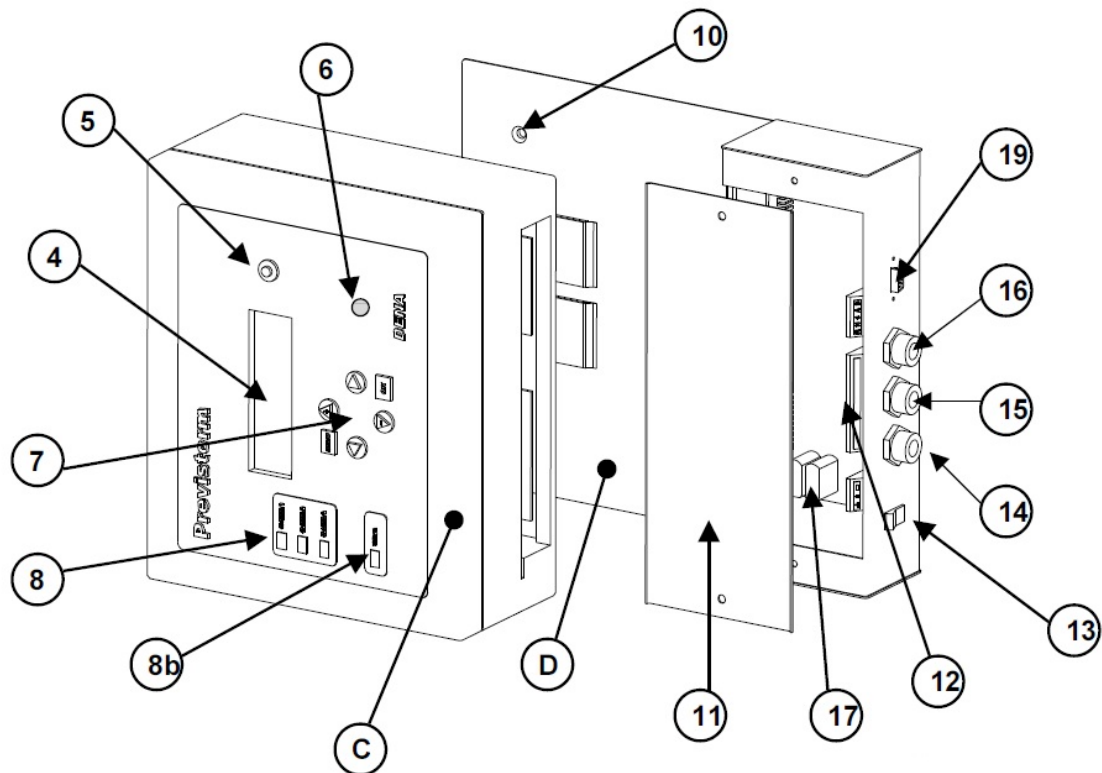
- e) **Evitar** el envejecimiento innecesario de los sistemas de protección.
- f) **Garantizar** el funcionamiento de los equipos más sensibles reduciendo costes en su mantenimiento. ⁽¹⁶⁾

Figura 22. (A) Sensor MCE ⁽⁹⁾



- Base ⁽¹⁾ para el conector
- Una paleta giratoria ⁽²⁾.
- Tornillo ⁽¹⁸⁾ para fijación a mástil de 50 mm de diámetro.
- Cable ^(B) del Sensor
- Clavija de conexión ⁽³⁾

Figura 23. (C) Módulo de adquisición – MAD ⁽⁹⁾



- Pantalla LCD ⁽⁴⁾
- Botón de iluminación ⁽⁵⁾ de la pantalla
- Zumbador ⁽⁶⁾
- Teclado de programación ⁽⁷⁾
- Indicadores ⁽⁸⁾ luminosos para cada alarma
- Indicador ^(8b) luminoso de error o fallo de sistema.
- Conector ⁽⁹⁾
- Platina ^(D) de la caja
- Agujeros ⁽¹⁰⁾ de fijación
- Cubierta ⁽¹¹⁾
- Bornes de conexión ⁽¹²⁾
- Interruptor ⁽¹³⁾ de marcha / paro
- Pasamuro del cable de alimentación ⁽¹⁴⁾
- Pasamuro para señales de alarma ⁽¹⁵⁾
- Pasamuro del cable del captador ⁽¹⁶⁾
- Puerto RS232 ⁽¹⁹⁾ para comunicación PC-MAD

Figura 24. Interconexión MCE-MAD ⁽⁹⁾

El cable de conexión entre la unidad central y el sensor MCE responde al siguiente esquema:

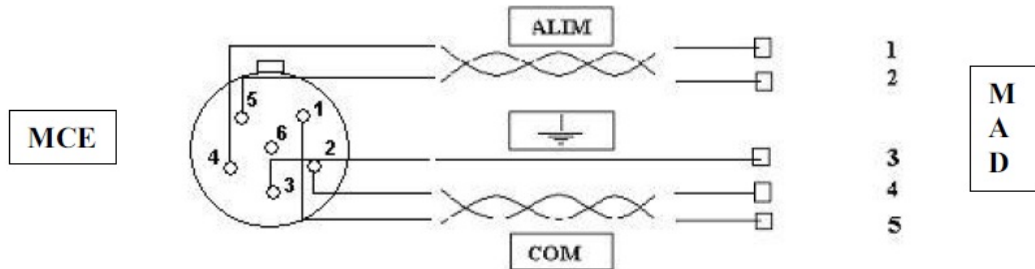
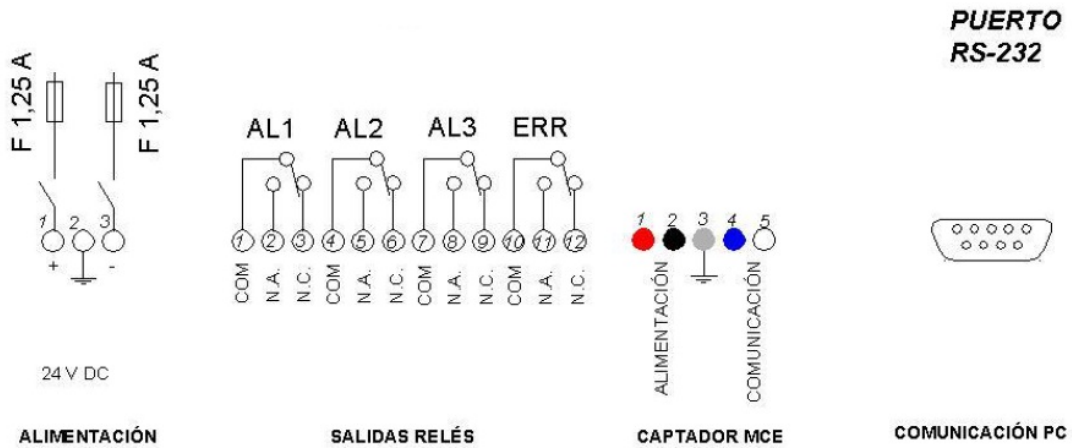


Figura 25. Puertos de alimentación, comunicación y alarma ⁽⁹⁾



Funcionamiento

El funcionamiento del equipo es totalmente automático. La puesta en marcha o la parada se efectúa mediante el interruptor ⁽¹³⁾. El hecho de apagar el equipo no borra la configuración ya establecida. ⁽⁹⁾

El campo electrostático se visualiza de forma permanente sobre la pantalla. Se puede activar la iluminación de la pantalla mediante la acción del pulsador ⁽⁵⁾. La iluminación de la pantalla desaparece al cabo de 10 minutos, de forma automática. ⁽⁹⁾

Cuando el valor de campo corregido sobrepasa alguno de los niveles de alarma, se activa el indicador luminoso de la correspondiente alarma y basculan los relés correspondientes. En la pantalla aparece un mensaje indicando el tipo de alarma activada. ⁽⁹⁾

En caso de fallo del Previstorm, el indicador luminoso de "Error" se enciende, y el relé de error bascula.

La configuración de los niveles y los tiempos de alarma se realizan mediante los pulsadores de configuración. Los niveles recomendados ya configurados en el equipo por defecto para la activación de alarmas son:

Tabla 11. Tipos de alarmas ⁽⁹⁾

Alarma 1	Alarma 2	Alarma 3
Tormenta Lejana	Tormenta Próxima	Tormenta Activa
2kV/m	4kV/m	8kV/m
Formación inicial de la tormenta, baja probabilidad de descargas a tierra.	Actividad entre nubes intensa. Mayor probabilidad de descargas a tierra.	Alta probabilidad de riesgo de caída de rayos a tierra.

El objetivo es prevenir a la población de la posible actividad eléctrica de una tormenta mediante la información de la probabilidad de caída de rayos.