



## INFORME DE FINAL DE PROYECTO

UNIVERSIDAD NACIONAL DE LA MATANZA  
DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS

Código: C160

Título del Proyecto: Aplicación de "Internet de la Cosas" al ahorro energético en establecimientos agrícolas

Programa de Investigación: PROINCE

Director del Proyecto: Lupi Oreste Daniel

Integrantes del Proyecto: Canziani, Mónica; Gomez, Rodrigo; Musa, Jonathan; Nassipián, Verónica; Slawiski Javier; Turconi, Diego; Castellini, Ariel; Lanzelotti, José Pablo; Oppedisano, Ruben; Zaradnik, Ignacio

Fecha de inicio: 2014/01/01

Fecha de finalización: 2015/12/31

Resumen:

El sector agroalimentario supone cerca del 30 por ciento del consumo mundial de energía, y más del 20 por ciento del total mundial de emisiones de gases de efecto invernadero. Ambos factores de gran importancia en la actualidad, el primero de ellos debido a la escasez de recursos energéticos que está sufriendo el país y el segundo de ellos relacionado con los aspectos ecológicos y medioambientales. En las explotaciones, de un modo general la energía sirve para bombear agua, criar el ganado, operar establecimientos avícolas y para crianza de cerdos, cultivar y recolectar, calentar los cultivos protegidos, el secado y el almacenamiento. Después de la recolección, se utiliza en el procesado, empaquetado, almacenamiento, transporte y consumo. Por otro lado es un objetivo fundamental del Gobierno Nacional el fomento del uso racional de la energía eléctrica, conforme los lineamientos regulatorios que se vienen sosteniendo desde el año 2004. Lo que trae asociado el desarrollo de nuevas tecnologías como ser dispositivos más eficientes, fuentes de energía alternativas y Redes Inteligentes (SMART GRIDS). Siendo estos último sistemas una de las herramientas que se están investigando para lograr dicho objetivo. El proyecto tiene como objetivo aplicar el concepto y técnicas propios de "Internet de la Cosas" al seguimiento de establecimientos agrícolas de modo tal de monitorear parámetros de proceso, ambientales y energéticos y combinarlos con información de geo-localización y meteorológica. Como resultado se podría eventualmente estimar la energía que va necesitar el establecimiento y de esta forma sugerir compartir las fuentes tradicionales de energía con fuentes alternativas y mantener los niveles de energía demandada a la red en un valor conveniente para todos.

Palabras claves: *Internet de las cosas, energías renovables, redes eléctricas inteligentes, establecimientos agropecuarios*

Área de conocimiento: Área de conocimiento: Ingeniería de comunicaciones, electrónica y control

Código de Área de Conocimiento: 1800

Disciplina: *Control*

Código de Disciplina: 1804

Campo de Aplicación: *Control*

Código de Campo de Aplicación: 1804

Otras dependencias de la UNLaM que intervinieron en el Proyecto: Lacie

Otras instituciones intervinientes en el Proyecto: INTA Castelar, INTI.

Otros proyectos con los que se relaciona:

## Aplicación de “Internet de la Cosas” al ahorro energético en establecimientos agrícolas

El sector agroalimentario -incluyendo la fabricación de insumos, producción, procesado, transporte, comercialización y consumo-, supone cerca del 30 por ciento del consumo mundial de energía, y más del 20 por ciento del total mundial de emisiones de gases de efecto invernadero.

Ambos factores de gran importancia en la actualidad, el primero de ellos debido a la escasez de recursos energéticos que está sufriendo el país y el segundo de ellos relacionado con los aspectos ecológicos y medioambientales. Además ambos aspectos están íntimamente relacionados, ya que una de las formas más comunes de generar energía es a través de la combustión de algún combustible de fósil como ser el gas o carbón, los cuales generan emisiones de gases invernadero.

En las explotaciones, de un modo general la energía sirve para bombear agua, criar el ganado, operar establecimientos avícolas y para crianza de cerdos, cultivar y recolectar, calentar los cultivos protegidos, el secado y el almacenamiento. Después de la recolección, se utiliza en el procesado, empaquetado, almacenamiento, transporte y consumo.

Por otro lado es un objetivo fundamental del Gobierno Nacional el fomento del uso racional de la energía eléctrica, conforme los lineamientos regulatorios que se vienen sosteniendo desde el año 2004, entre los que podemos citar GENREN, PERMER, FONARSEC, PRONUREE, etc. Lo que trae asociado el desarrollo de nuevas tecnologías como ser dispositivos más eficientes, fuentes de energía alternativas y Redes Inteligentes (SMART GRIDS). Siendo estos último sistemas una de las herramientas que se están investigando para lograr dicho objetivo.

El proyecto tiene como objetivo aplicar el concepto y técnicas propios de “Internet de la Cosas” al seguimiento de establecimientos agrícolas de modo tal de monitorear parámetros de proceso, ambientales y energéticos y combinarlos con información de geo-localización y meteorológica. Como resultado del manejo de esta información se podría eventualmente estimar la energía que va necesitar el establecimiento y de esta forma sugerir compartir las fuentes tradicionales de energía con fuentes alternativas como ser biogás, fotovoltaica, eólica, solar térmica, etc. y mantener los niveles de energía demandada a la red en un valor conveniente para todos.

Para la realización del proyecto de partirá de las estaciones desarrolladas por nuestro laboratorio para el proyecto “Sistema inalámbrico de microsensores para aplicaciones agropecuarias”. Las mismas serán ampliadas en su capacidad para sensar parámetros energéticos adicionales tales como energía eléctrica, volumen de gas consumido, salto térmico, gradiente térmico en paredes y otros propios de cada actividad.

Se estudiarán en una primera etapa los establecimientos agrícolas con mejor posibilidad de intervención, en acuerdo con colegas del INTA consultados, por ejemplo avicultura y lechería, pero se puede extender a otros casos.

Igualmente se estudiarán las fuentes alternativas de energía disponibles, desde el punto de vista de su aplicación y posibilidades de monitoreo remoto.

Se desarrollarán los sensores para las mediciones de parámetros relacionados con el consumo energético y su interface con las estaciones inalámbricas disponibles, pero adaptadas de modo tal que utilicen un protocolo adecuado para su comunicación directa vía internet.

Se plantea desarrollar una instalación piloto de monitoreo mediante internet de las cosas en un establecimiento pequeña o mediana envergadura, de acuerdo con un convenio a firmarse con INTA a tal efecto.

Como conclusión del proyecto, se espera alcanzar experiencias suficientes para recomendar políticas y estrategias que apoyen el desarrollo y aplicación del concepto de "Internet de la Cosas" para la optimización y utilización de los recursos renovables

Otro objetivo buscado es consolidar el conocimiento y fortalecer los recursos humanos en lo que respecta al manejo y aplicación de estas tecnologías informáticas, en un área tan sensible a las necesidades de nuestro País.

Palabras claves: *Internet de las cosas, energías renovables, redes eléctricas inteligentes, establecimientos agropecuarios*

## Contenido

1- Introducción .....	10
1.1- La situación energética en Argentina .....	10
1.2- Fuentes de energía renovables .....	15
1.2.1 - Energía eólica .....	16
1.2.2- Energía geotérmica .....	16
1.2.3- Energía solar .....	16
1.2.3.1- Energía solar fototérmica .....	17
1.2.3.2- Energía solar fotovoltaica .....	17
1.2.3.3- Energía solar termoeléctrica .....	17
1.2.3.4- Energía solar - Conclusión .....	17
1.2.4- Energía de la biomasa .....	18
1.2.5- Energía de biocarburantes o biocombustibles .....	18
1.2.6- Energía marina .....	18
1.2.7- Energía hidráulica .....	19
1.3 - Redes Eléctricas Inteligentes (REI) o Smart Grids .....	19
1.4- Medidas a desarrollar para mejorar la eficiencia energética en la industria agropecuaria .....	22
1.5- Soluciones tecnológicas para mejorar la eficiencia energética .....	27
1.6- Alcance del proyecto .....	31
2- Microestaciones .....	32
2.1- Etapa de Fuente de Alimentación .....	32
2.1.1- Montaje .....	32
2.1.2- Detalle del Proceso de Prueba .....	33
2.2- Etapa de Sensores y Transceptor .....	33
2.2.1- Montaje .....	33
2.2.2- Proceso de Prueba de los Sensores .....	35
2.3- Lectura de los sensores .....	36
2.3.1- Resumen de sensores utilizados en la Microestación .....	37
2.3.2- Aspectos Preliminares .....	37
2.3.3- Sensor de temperatura AT30TSE75 .....	38
2.3.3.1- Dirección I2C del Sensor .....	38
2.3.3.2- Descripción del Registro de Control .....	38
2.3.3.3- Descripción del Registro de Datos .....	39
2.3.3.4- Proceso de Lectura .....	39
2.3.3.4.1- Rutina de Inicialización del Sensor de Temperatura AT30TS75 .....	39
2.3.3.4.2- Rutina de lectura del Sensor de Temperatura AT30TS75 .....	40
2.3.4- Sensor de Presión Atmosférica MPL3115A2 .....	41
2.3.4.1- Dirección I2C del Sensor .....	41
2.3.4.2- Descripción de los Registros de Control .....	41
2.3.4.3- Descripción de los Registros de Datos .....	43
2.3.4.4- Proceso de Lectura .....	44
2.3.4.5- Rutina de Inicialización del Sensor de Presión Atmosférica MPL3115A2 .....	46
2.3.4.6- Rutina de Activación del Sensor de Presión Atmosférica MPL3115A2 .....	47
2.3.4.7- Rutina de Lectura del Sensor de Presión Atmosférica MPL3115A2 .....	48
2.3.5- Sensor de luz ambiente BH1721FVC .....	48
2.3.5.1- Inicialización .....	48
2.3.5.2- Función de configuración inicial .....	48
2.3.5.3- Lectura de la medición .....	49
2.3.5.4- Conversión de la medición a Lux .....	49
2.3.5.5- Función de lectura de la medición .....	49
2.3.6- Sensor de Humedad Relativa y Temperatura Si7005 .....	49
2.3.6.1- Dirección I2C del Sensor .....	49
2.3.6.2- Descripción del Registro de Control .....	49
2.3.6.3- Descripción de los Registros de Datos .....	50

2.3.6.3.2- Lectura de Temperatura.....	51
2.3.6.4- Proceso de Lectura de la Humedad Relativa .....	52
2.3.6.5- Proceso de Lectura de la Temperatura .....	53
2.3.6.6- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Configuración Para Lectura de Humedad .....	54
2.3.6.7- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Configuración Para Lectura de Temperatura .....	55
2.3.6.8- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Lectura de Humedad .....	55
2.3.6.9- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Lectura de Temperatura .....	56
2.4- Conclusiones .....	56
3- Transceptor de la microestación .....	57
3.1- Tipos de protocolos de comunicación .....	57
3.1.1-Lightweight Mesh .....	57
3.1.2- BitCloud .....	59
3.1.3- Contiki.....	60
3.1.4- Selección del stack .....	61
3.2- Pruebas de la etapa de RF .....	61
3.3- Presentación general de la aplicación p2p.....	63
3.4- Enlace punto a punto Bit Cloud(P2P-BC) .....	67
3.5- WSN Demo.....	78
3.6- Contiki – 6LowPan.....	85
3.7- Conclusión.....	85
4- Estación colectora .....	86
4.1- Descripción General – Hardware .....	86
4.2- Firmware.....	86
4.2.1- Descripción general .....	87
4.2.2- Herramientas y entorno de trabajo .....	87
4.2.3- Proyecto Base .....	88
4.2.4- Nuevo planteo.....	89
4.3 – Protocolo microestación.....	89
4.3.1 –Detalle de la Trama de datos .....	89
4.3.2 –Detalle de la Trama de comandos.....	90
4.4- Desarrollo .....	91
4.5- Pruebas y ensayos. ....	112
4.6 –Conclusiones .....	117
5- Software .....	119
5.1- Tipos de arquitecturas de comunicación.....	119
5.1.1- Esquema de la Arquitectura por Puerto Serie RS232 .....	119
5.1.2- Esquema de la Comunicación mediante un Servidor de Correo .....	120
5.1.3- Esquema de la Arquitectura para TCP/IP.....	120
5.2- Tipos de Tramas.....	121
5.2.1- Detalle de la Trama de datos .....	122
5.2.2- Detalle de la Trama de comando .....	123
5.3- Cliente final <C160client.exe> .....	123
5.3.1- Estudio de Mercado .....	123
5.3.1.1- Metzoo .....	124
5.3.1.2- Booster (AgruWeather) .....	125
5.3.1.3- View Point.....	127
5.3.1.4- Ubidots.....	128
5.3.1.5- Digi Device Cloud .....	130
5.3.1.6- Telit.....	131
5.3.1.7- Resumen del Estudio de Mercado .....	132
5.3.2- Planteo general del software a desarrollar .....	133

5.3.3- Comunicaciones .....	134
5.3.3.1- Comunicación local por puerto serie RS232 .....	134
5.3.3.2- Comunicación remota por correo electrónico .....	134
5.3.3.3- Comunicación remota por TCP/IP.....	134
5.3.4- Descripción de la aplicación cliente.....	134
5.3.4.1- Ingreso al sistema.....	134
5.3.4.2- Pantalla principal.....	135
5.3.4.3- Inicio/Cierre de sesión dentro del sistema.....	136
5.3.4.4- Establecimientos.....	136
5.3.4.5- Ver establecimientos y Sensores .....	136
5.3.4.6- Agregar nuevo establecimiento.....	136
5.3.4.7- Borrar establecimiento .....	136
5.3.4.8- Modificar Sensores .....	138
5.3.4.9- Gestión de Usuarios.....	138
5.3.4.10- Cambiar contraseña.....	139
5.3.4.11- Alta nuevo usuario .....	139
5.3.4.12- Baja usuario existente.....	140
5.3.4.13- Modificación de privilegios .....	140
5.3.4.14- Conexiones.....	140
5.3.4.15- Configuración TCP/IP .....	141
5.3.4.16- Configuración Puerto Serie RS232 .....	141
5.3.4.17- Ayuda.....	141
5.3.4.18- Botones panel izquierdo Pantalla principal.....	142
5.3.4.19- Ver en mapa .....	142
5.3.4.20- Datos Históricos.....	143
5.3.4.21- Tendencias .....	144
5.4- Servidor de datos <C160server.exe>.....	146
5.4.1- Software intérprete.....	146
5.4.2- Motor de base de datos .....	148
5.4.2.1- Paradox y BDE (Borland Database Engine).....	148
5.4.2.2- MySQL.....	149
5.4.2.3- Oracle .....	149
5.4.2.4- Selección de motor de base de datos .....	150
5.4.3- Implementación de base de datos.....	150
5.4.3.1- Oracle .....	150
5.4.3.1.1- Modelo de base de datos.....	150
5.4.3.1.2- Pasos a realizar .....	150
5.4.3.1.3- Conexión a Base de Datos mediante ODBC .....	150
5.4.3.1.4- Conexión desde el proyecto software con base de datos.....	153
5.4.3.1.5- Modelo de consulta SQL básica.....	155
5.4.3.1.6- Cambio de motor de base de datos .....	155
5.4.3.2- Paradox .....	156
5.4.3.2.1- Modelo de base de datos.....	156
5.4.3.2.2- Pasos realizados.....	156
5.4.3.3- Nuevo modelo de datos. ....	157
5.4.3.3.1- Descripción de tablas.....	158
5.4.3.3.1.1- Tabla USUARIO.....	158
5.4.3.3.1.2- Tabla ESTABLECIMIENTO.....	158
5.4.3.3.1.3- Tabla CONCENTRADOR.....	158
5.4.3.3.1.4- Tabla ESTACION.....	159
5.4.3.3.1.5- Tabla SUBESTACION.....	159
5.4.3.3.1.6- Tabla PRIVILEGIO.....	159
5.4.3.3.1.7- Tabla SENSOR.....	159
5.5- Simulador para pruebas y desarrollo <C160sim.exe>.....	160
6- Pruebas del Sistemas.....	161

6.1- Descripción de la Demo.....	161
6.2- Sistemas de generación de energía renovable.....	165
7- Medición de Energía eléctrica.....	166
7.1- Parámetros de medición.....	166
7.2- Tipos de medidores.....	166
7.3- Planteo inicial.....	168
8.1- Introducción.....	170
8.2 - Descripción del sistema.....	170
8.2.1- Paneles solares.....	171
8.2.2 - Seguimiento solar.....	172
8.2.3 - Baterías (electricidad).....	172
8.2.3.1 - Principios de funcionamiento.....	172
8.2.3.2 - Tipos de batería.....	173
8.2.3.2.1- Batería AGM.....	173
8.2.3.2.2- GEL.....	173
8.2.3.2.3- MONOBLOCK.....	173
8.2.3.2.4- OpzS.....	173
8.2.4 - LEDs.....	174
8.3 - Estudio de mercado.....	174
8.3.1- Necesidad del mercado.....	175
8.4 - Desarrollo.....	176
8.4.1 - Elección del panel solar.....	177
8.4.2 - Estimación del consumo.....	178
8.5 - Dimensionado del generador fotovoltaico.....	179
8.5.1 - Dimensionado del sistema de acumulación.....	179
8.5.2 - Dimensionado del regulador.....	181
8.5.3 - Regulador elegido [62].....	183
8.6 - Sensores utilizados en el seguimiento solar.....	185
8.7 - Elección del Microcontrolador.....	186
8.8 - Motores utilizados.....	186
8.8.1 - Controlador de motores.....	187
8.9 - Medición de tensión, posición vertical y posición horizontal.....	188
8.9.1 - Medición de tensión.....	188
8.9.2- Medición de posición vertical y horizontal.....	189
8.10 - Soporte mecánico del panel solar.....	189
8.11 – Firmware.....	190
8.12- Esquemáticos y PCB.....	193
8.12.1- Circuito placa microcontrolado.....	193
8.12.2-Circuito puente H.....	194
8.12.3- PCB placa micro.....	194
8.12.4- PCB del puente H.....	195
8.13- Evaluación.....	195
8.13.1 - Eficiencia.....	195
8.13.2 - Sistema sin seguidor (fijo):.....	195
8.13.3 - Sistema con seguidor:.....	197
8.13.4 - Comparación entre sistemas:.....	197
8.13.5 - Precisión.....	198
8.14- Conclusiones.....	201
9 - Biodigestor.....	203
9.1 - Introducción.....	203
9.2 - Biodigestores en Argentina.....	204
9.3 - Introducción teórica.....	205
9.3.1 - El proceso.....	205
9.3.2 - Factores influyentes.....	206
9.4 - Materiales.....	208

9.5 - Clasificación de Biodigestores .....	210
9.5.1 - Biodigestor de carga discontinua o intermitente (Tipo Batch).....	211
9.5.2 - Biodigestor semi-continuo.....	211
9.5.3 - Biodigestor Continuo.....	212
9.5.3.1 - Desplazamiento Horizontal .....	213
9.6 - Estudio de mercado.....	213
9.7 - Definición del producto .....	215
9.7 - Desarrollo .....	216
9.8 - Diseño mecánico .....	217
9.9 - Microcontrolador .....	220
9.10 - Sensor de Temperatura .....	220
9.11 - Sensor de Presión .....	221
9.12 - Resistencia .....	221
9.13 - Esquemáticos, PCB y distribución de los componentes.....	222
9.14 - Firmware.....	224
9.15 - Visualización.....	226
9.16 - Ensayo de los Sensores .....	226
9.17 Conclusiones .....	229
9.18 - Mejoras a futuro.....	230
10- Generador Eólico de baja potencia.....	231
10.1- Introducción .....	231
10.2- Energía eólica en argentina .....	231
10.3- Generadores eólicos.....	233
10.4- Dimensionamiento del aerogenerador. ....	234
10.5- Estudio de mercado.....	234
10.5.1- Definición de la potencia del aerogenerador. ....	235
10.6- Principio básico de funcionamiento de aerogenerador modelo Hugh Piggott... 235	
10.6.1- Descripción mecánica. ....	235
10.7- Desarrollo .....	238
10.7.1- Bloques y conexionado del sistema .....	238
10.7.1.1- Conexionado general del sistema .....	238
10.7.1.2- Rectificador.....	238
10.7.1.3- Fusibles .....	239
10.7.1.4- Freno del molino .....	239
10.7.1.5- Selector o Controlador de Carga y Resistor de Descarga .....	239
10.7.1.6- Baterías .....	239
10.7.1.7- Inversor.....	240
10.7.1.8- Jabalina .....	240
10.7.2- Aspas, pivote y rotor .....	240
10.7.3- Estator .....	240
10.7.3.1- Selección de las bobinas.....	241
10.7.3.2- Construcción de las bobinas .....	241
10.7.3.3- Cálculo teórico de las bobinas .....	241
10.7.3.3- Conexionado de las bobinas .....	243
10.7.3.4- Protección de las bobinas .....	245
10.7.3.5- Mediciones sobre las bobinas .....	246
10.7.4- Controlador de Carga.....	247
10.7.4.1- Diseño esquemático.....	248
10.7.4.2- Principio de funcionamiento .....	248
10.8- Ensayos y experimentos.....	248
10.8.1- Ensayo de rendimiento .....	249
10.8.2- Ensayo de plegado .....	249
10.8.3- Ensayo de posicionamiento .....	249
10.8.4- Pruebas realizadas .....	249
10.8.5- Otras posibles pruebas .....	251

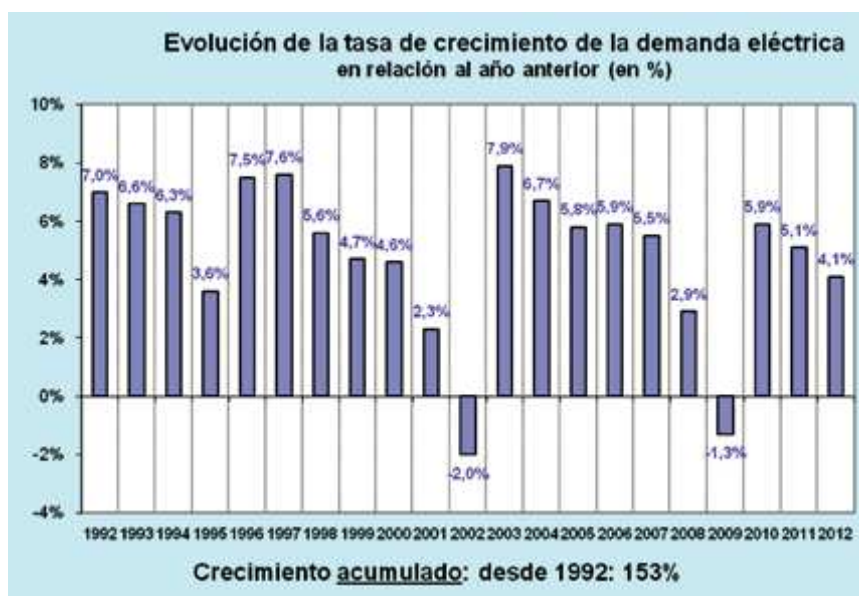


10.9- Conclusiones .....	252
11 - Sistema de transferencia automática para grupos electrógenos .....	253
11.1 - Introducción .....	253
11.2 - Estudio de Mercado .....	253
11.3 - Descripción general .....	254
11. 4 - Hardware .....	255
11.5 - Entorno de desarrollo integrado(IDE).....	255
11.6 - Firmware.....	256
11.7 – Ensayos .....	257
11.8 - Conclusiones .....	258
12 - Evaluación del equipo y presupuesto.....	261
12.1- Periodo 2014. ....	261
12.2- Periodo 2015. ....	261
13 - Transferencias.....	261
13.1- Periodo 2014. ....	261
13.2- Periodo 2015. ....	262
14- Agradecimientos .....	262
15 – Conclusiones .....	262
16 - Bibliografía.....	263

## 1- Introducción

### 1.1- La situación energética en Argentina

En Argentina, como en todo Latinoamérica, la situación energética presenta un momento de quiebre por ser una economía en desarrollo, se enfrenta a una creciente demanda energética que corresponde a un incremento de las producciones industriales y a una mejora de la calidad de vida. Todo crecimiento económico del país repercute en una mayor demanda de energía. Esto se puede ver en los datos de consume de energía eléctrica publicados mensualmente por Fundelec, los cuales son reproducidos por múltiples medios digitales e impresos. En los últimos dos años la revista Ingeniería Eléctrica ha publicado un balance de los consumos eléctricos del año anterior, es decir un balance de los años 2012 y 2013 [14][15]. Los cuales muestran el crecimiento en la demanda año a año. El grafico N°1.1 muestra la evolución del consumo desde 1992 y presenta un crecimiento acumulado de un 153% en 20 años [14]. No siendo excepción el año 2013 con un crecimiento del 3.45% [15].



**Grafico N°1.1- Evolución del consumo de energía eléctrica.**

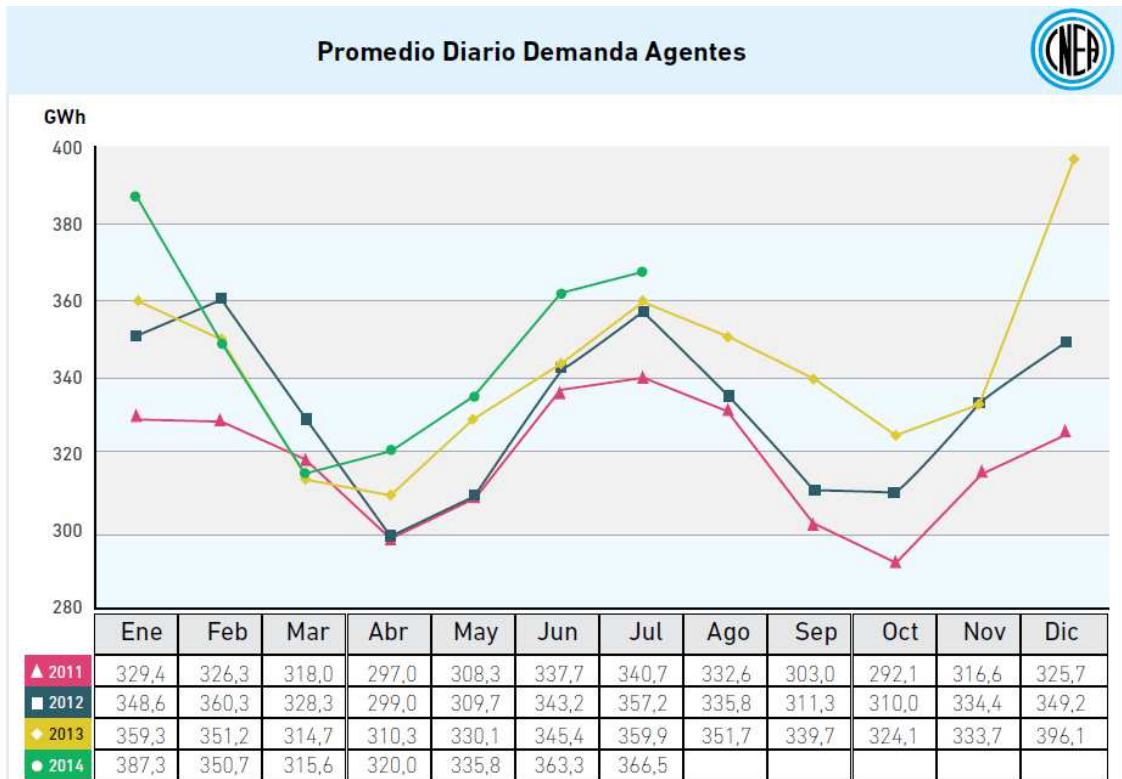
Mayor información sobre consumos y generación de energía eléctrica se puede encontrar en la síntesis del mercado eléctrico mayorista de la República Argentina, que mensualmente, desde mayo del 2003, la Comisión Nacional de Energía Atómica elabora [7].

Según los informes brindados por éste ente en el mes de Julio del 2014, las variaciones de la demanda neta fueron las siguientes:

- La variación “mensual” +1,8 %
- El “año móvil” +4,4%
- El “acumulado anual” +2,9%

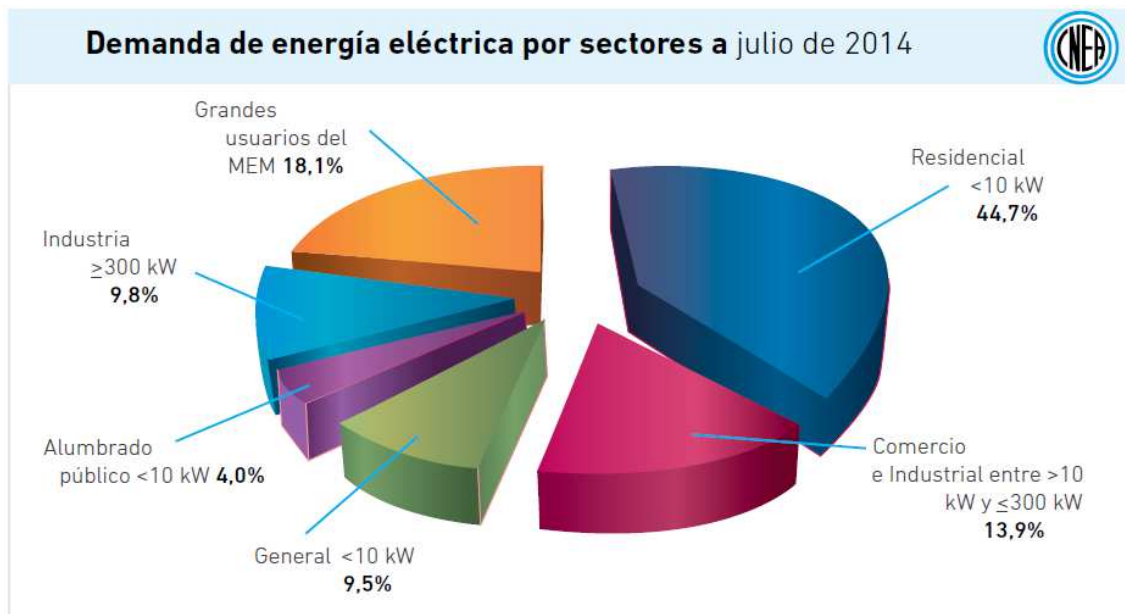
La variación “mensual” se calcula computando la demanda neta de los agentes, sin considerar las pérdidas en la red en alta tensión, respecto del mismo valor del año anterior. El “año móvil” compara la demanda de los últimos 12 meses respecto de los 12 anteriores. El “acumulado anual”, en cambio, computa los meses corridos del año en curso, respecto de los mismos meses del año pasado.

En cuanto al promedio diario de la demanda agentes, este mes se registró un crecimiento del 1,8%, en comparación con los datos de julio de 2013. El valor registrado es además el mayor valor de los últimos cuatro años para el mes de julio, como se puede ver en el grafico N°1.2.



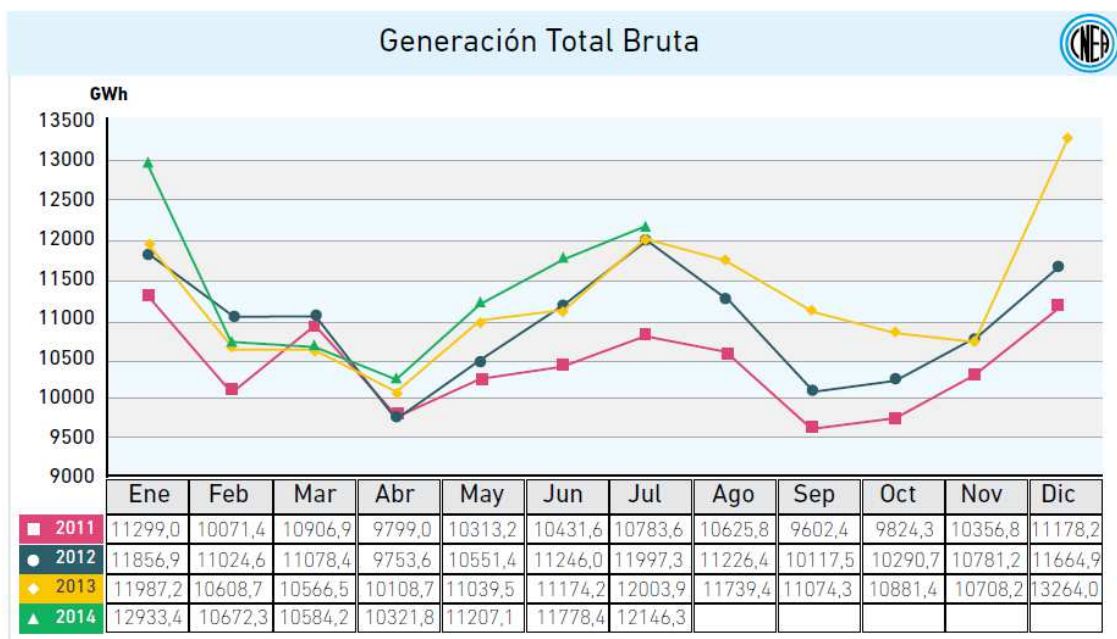
**Grafico N°1.2 - Promedio diario de demanda de agentes**

El grafico N°1.3 presenta la demanda de energía eléctrica por sectores.



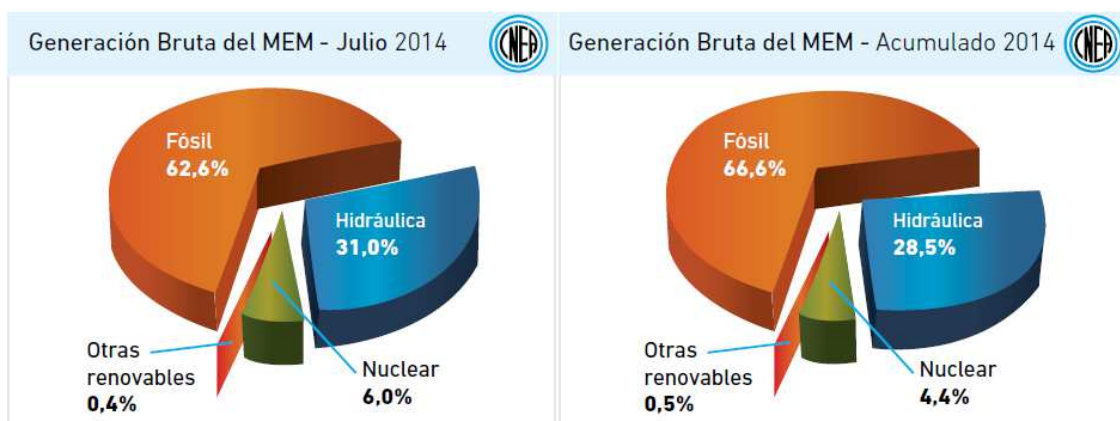
**Grafico N°1.3 - Demanda de energía eléctrica por sectores**

A su vez la generación bruta nacional vinculada al SADI (Sistema Argentino de Interconexión) fue un 1,2% superior a la de julio del 2013. El valor registrado es además el mayor de los últimos cuatro años para el mes de julio, grafico N°1.4.



**Grafico N°1.4 - Generación bruta de energía.**

El grafico N°1.5 se presenta la relación entre las distintas fuentes de generación. La generación de energías no renovables en nuestro país es aún no muy elevada comparada con las energías tradicionales.



**Gráfico N°1.5 - Fuentes de generación de energías**

La generación de “otras renovables”, que surge de las gráficas precedentes, comprende la generación eólica y fotovoltaica incorporada hasta el momento. Cabe destacar que el mayor porcentaje de dicho valor corresponde a la generación eólica.

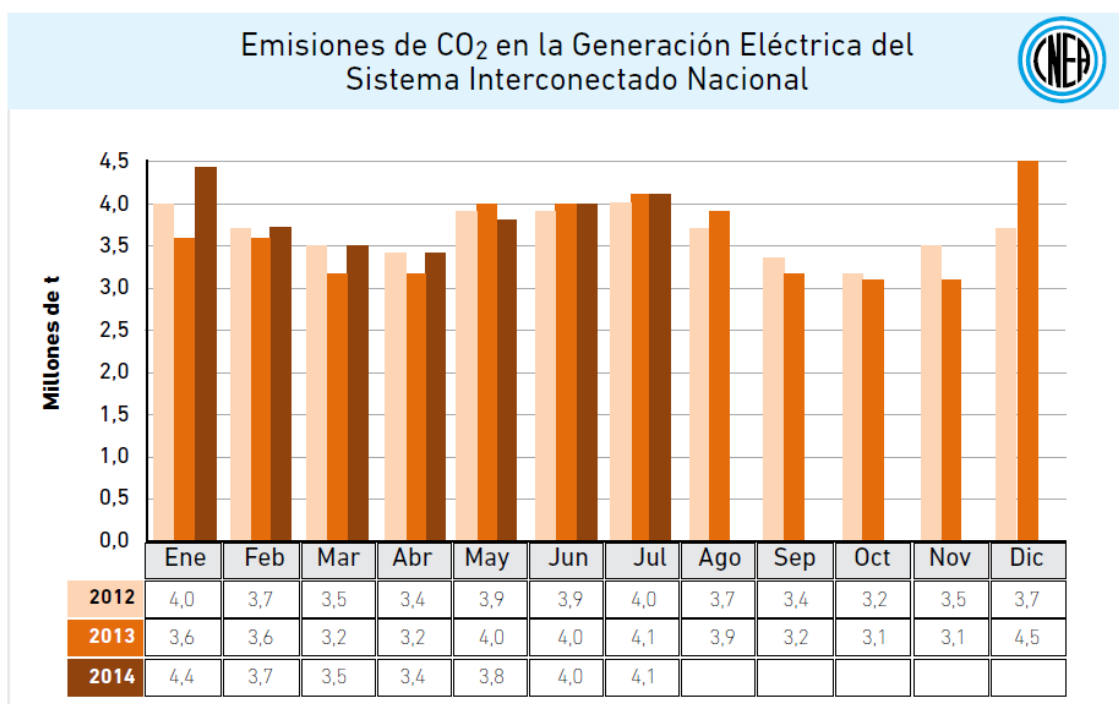
Se puede observar que la principal fuente de energía explotada en nuestro país es la energía fósil (62,6 % de lo consumido en Arg.) donde allí entran el uso de combustibles de tipo no renovables como ser el gas, petróleo, fuel oil y carbón.

La segunda fuente de energía más utilizada es la energía hidráulica (31 %) siendo una fuente del tipo renovable que aprovecha la energía potencial que posee la masa de agua en virtud de un desnivel.

En mucho menor medida se encuentran el uso de la energía nuclear (6%) y las generadoras de fuentes alternativas eólicas y fotovoltaicas aportaron (0,4%).

Asociada a las fuentes actuales de generación, principalmente a la fósil, existe una generación de contaminación.

En el gráfico N°1.6, se muestran las emisiones de CO<sub>2</sub>, derivadas de la quema de combustibles fósiles en los equipos generadores vinculados al MEM (Mercado Eléctrico Mayorista) durante los últimos 3 años, en millones de toneladas.



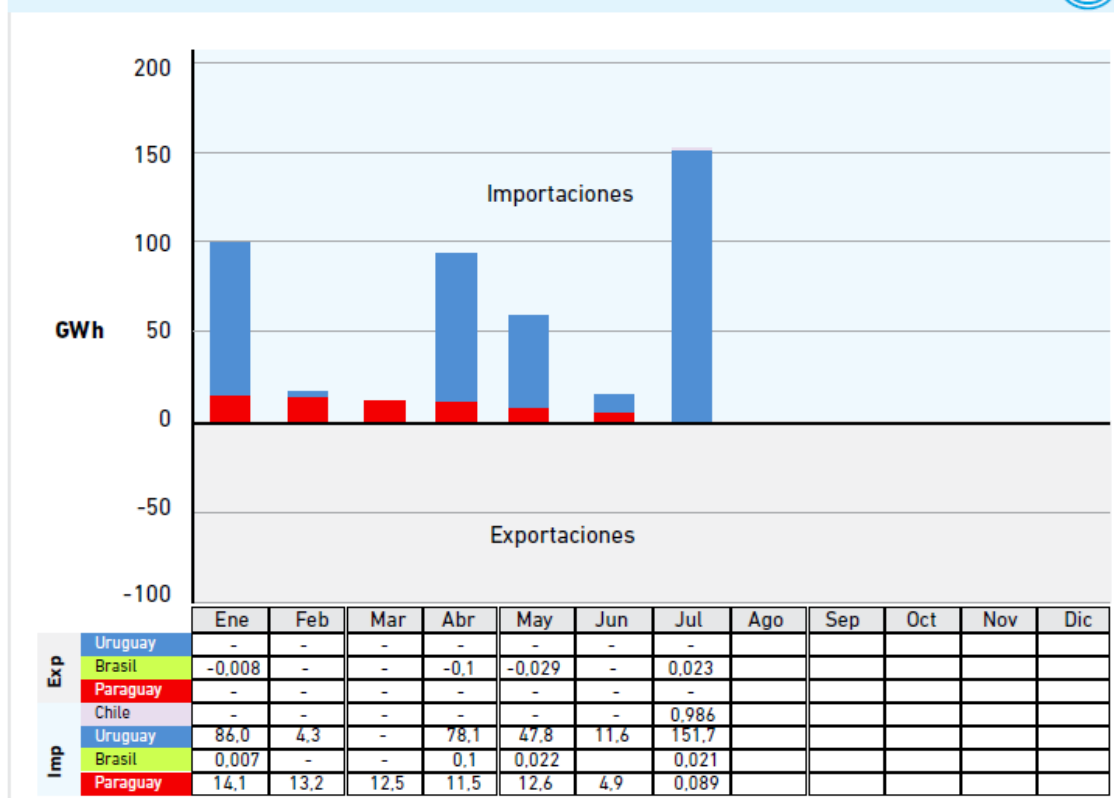
**Gráfico N°1.6 - Emisiones de CO<sub>2</sub> en la generación eléctrica del Sistema Interconectado Nacional**

Debido a ello es que se necesita un mayor desarrollo en la investigación del uso de todo tipo de energías renovables o las también llamadas energías limpias.

Además de la energía generada por distintos medios, el país importa energía de los países limítrofes, en el grafico N°1.7 se ve la evolución que han tenido las importaciones y exportaciones de energía con Brasil, Paraguay y Uruguay, en GWh durante los meses corridos del año 2014.

La importación en el mes de julio representó aproximadamente el 1,25% de la demanda total.

## Evolución Importaciones/Exportaciones 2014



**Gráfico N°1.7 - Evolución de importaciones y exportaciones 2014**

Si bien la importación de energía no parece ser un gran porcentaje, no debemos olvidarnos que para la generación de la misma se está importando una gran cantidad de combustible de origen fósil, importaciones las cuales los últimos años han ido en aumento [27][28][29]. El Tabla N°1.1 nos muestra las exportaciones e importaciones en millones de dólares, según Grandes rubros y Usos económicos y variación porcentual de Julio y siete meses 2013-2014. Donde se puede ver que para el mes de Julio del 2014 se registro una importación de 1.270 millones de dólares en combustibles y lubricantes [30], así como un acumulado de los últimos 7 meses de 7.226 millones de dólares.

Es por estos motivos que se plantean dos caminos complementarios y claramente diferenciados: por un lado, el cambio de conducta para evitar el consumo innecesario, y por otro la investigación y el aprovechamiento de energías sustentables que permitan acompañar el crecimiento del país y el aumento de la demanda del suministro.

Como se pudo ver en el gráfico N°1.5 la Argentina avanza en a paso lento en este tema. La producción de energía a partir de fuentes renovables representa en el mes de julio solo el 0,4% del total. A pesar de que desde 1998 que existen marcos jurídicos de apoyo y promoción a las energías renovables en el país , que en 2006, se sancionó la Ley Nacional 26.190 para complementar la Ley 25.019 y con la particularidad de incluir a las energías eólica, solar, geotérmica, mareomotriz, hidráulica, la biomasa y el biogás de diferentes orígenes [33]. Esta Ley fijó un objetivo claro: para el 2016, el 8% del consumo eléctrico local deberá ser abastecido con fuentes de energías renovables.

Usos	Julio			Siete meses		
	2013*	2014*	Variación porcentual	2013*	2014*	Variación porcentual
	Millones de U\$S		%	Millones de U\$S		%
<b>Total general</b>	<b>7.022</b>	<b>5.920</b>	<b>-16</b>	<b>42.797</b>	<b>39.014</b>	<b>-9</b>
<b>Bienes de capital</b>	<b>1.086</b>	<b>1.156</b>	<b>6</b>	<b>6.638</b>	<b>7.063</b>	<b>6</b>
Bienes de capital excluido equipos de transporte	829	849	2	5.244	5.344	2
Equipos de transporte industriales	257	308	20	1.393	1.719	23
<b>Bienes intermedios</b>	<b>1.774</b>	<b>1.592</b>	<b>-10</b>	<b>11.307</b>	<b>10.854</b>	<b>-4</b>
Suministros industriales elaborados no especificados en otras partidas	1.736	1.559	-10	11.045	10.661	-3
Alimentos y bebidas elaborados fundamentalmente para la industria	38	34	-11	262	192	-27
<b>Combustibles y lubricantes</b>	<b>1.358</b>	<b>1.270</b>	<b>-6</b>	<b>7.462</b>	<b>7.226</b>	<b>-3</b>
<b>Piezas y accesorios para bienes de capital</b>	<b>1.450</b>	<b>1.071</b>	<b>-26</b>	<b>9.130</b>	<b>7.585</b>	<b>-17</b>
<b>Bienes de consumo</b>	<b>672</b>	<b>559</b>	<b>-17</b>	<b>4.290</b>	<b>3.846</b>	<b>-10</b>
Artículos de consumo duraderos no especificados en otra partida	62	51	-18	417	362	-13
Artículos de consumo semiduraderos no especificados en otra partida	179	152	-15	1.140	984	-14
Artículos de consumo no duraderos no especificados en otra partida	282	237	-16	1.736	1.656	-5
Alimentos y bebidas básicos fundamentalmente para el hogar	24	28	17	175	195	11
Alimentos y bebidas elaborados fundamentalmente para el hogar	67	65	-3	483	425	-12
Equipos de transporte no industriales	57	25	-56	340	223	-34
<b>Vehículos automotores de pasajeros</b>	<b>666</b>	<b>252</b>	<b>-62</b>	<b>3.858</b>	<b>2.314</b>	<b>-40</b>
<b>Resto</b>	<b>17</b>	<b>20</b>	<b>18</b>	<b>113</b>	<b>128</b>	<b>13</b>

**Tabla N°1.1- Exportaciones e importaciones en millones de dólares.**

Algunos de los proyectos de Argentina para el uso de energías renovables son:

- La primera planta solar fotovoltaica piloto en Ullum, provincia de San Juan, conectada a la red del sistema eléctrico nacional, con un total de 1.2MW [1] [2].
- Se ha diseñado el proyecto y se obtuvieron los fondos para la realización del Parque Solar Cañada Honda III, que aportará una generación de 20MW de potencia al sistema interconectado nacional [3] [4].
- La Cooperativa Eléctrica Zapala y el INTI se proponen recuperar una minicentral hidroeléctrica para la generación de energía económica y sustentable, destinada al desarrollo de la producción local.
- En la región de la Patagonia abundan el viento y el agua, el plan es recuperar este tipo de fuentes de energía, el INTI, a través del Programa de Área Estratégica de Industrias de Servicios y Ambiente, la Coordinación Cuyo y el Centro Neuquén, realizó un estudio de prefactibilidad con el objetivo de determinar la factibilidad técnica y económica de la rehabilitación de la minicentral hidroeléctrica Covunco.

Nuestras sociedades modernas ya no pueden concebirse sin electricidad y cada vez más dispositivos que llegan para facilitarnos las tareas dependen de ella. A futuro, esta situación no parece revertirse. Para que esta realidad no perjudique a nuestro planeta y a la vez se satisfagan todas nuestras necesidades, es imperioso que se reflexione sobre el uso de todo tipo de energías, nuevas formas de generación y distribución menos perjudiciales que permitan el cuidado del medio ambiente que habitamos.

## 1.2- Fuentes de energía renovables

Teniendo en cuenta la necesidad de cuidar los recursos energéticos naturales y la conservación del medio ambiente mencionaremos los distintos tipos de energías renovables, algunas de las cuales profundizaremos más durante el desarrollo del proyecto.

Se denomina **energía renovable** a la energía que se obtiene de fuentes naturales virtualmente inagotables, ya sea por la inmensa cantidad de energía que contienen, o porque son capaces de regenerarse por medios naturales. Entre las energías

renovables se cuentan la eólica, geotérmica, hidráulica, mareomotriz, solar, la biomasa y los biocombustibles [17].

### **1.2.1 - Energía eólica**

Es el aprovechamiento de la energía contenida en el viento que puede ser transformada, según sea la necesidad, en energía eléctrica, mecánica o térmica.

Las posibilidades de uso que ofrece la energía eléctrica son bien conocidas, se destina, a satisfacer necesidades de trabajo y confort requeridas por el hombre. En cuanto a la mecánica, se utiliza para el bombeo de agua o molienda de distintos productos. La energía térmica se consigue a partir de la energía mecánica. Para efectuar esa transformación se utilizan distintos tipos de equipamientos. En términos generales no se requieren grandes velocidades de viento para producir energía, más bien al contrario, cuando el viento es demasiado intenso se hace necesario detener los equipos para evitar deterioro. En la mayoría de los casos, un equipo comienza a generar energía con una velocidad del viento de 4 metros por segundo (m/s), equivalente a unos 15 km/h. Entrega su potencia máxima cuando la velocidad es del orden de los 12 a 15m/s (40 a 55 km/h) y es necesario sacarla de servicio cuando alcanza 25m/s (90km/h) [16].

Para la generación de este tipo de energía se utilizan los aerogeneradores los cuales mueven una turbina y consiguen transformar la energía cinética del viento por energía mecánica.

La cantidad de energía que se puede obtener está en función del tamaño del "molino". A mayor longitud de las aspas, se obtiene más potencia y por lo tanto más energía. El tamaño de estos modernos molinos de viento suele variar, ya que existen unidades que van desde los 400 W y un diámetro de 3 metros. Utilizados para el uso doméstico de las casas, hasta los aerogeneradores comerciales instalados por empresas de electricidad que llegan a tener una potencia de 2,5 MW y 80 m. de diámetro de aspas.

### **1.2.2- Energía geotérmica**

Se entiende por energía geotérmica a aquella que, aprovechando el calor que se puede extraer de la corteza terrestre, se transforma en energía eléctrica o en calor para uso humano o procesos industriales o agrícolas. La Tierra almacena en forma de calor gran cantidad de energía. Diferentes hipótesis tratan de explicar a que se deben estas altas temperaturas existentes sobre el origen y posterior evolución del planeta. Las explicaciones más convincentes atribuyen a la acción combinada de varios fenómenos naturales, entre los que adquieren especial importancia los efectos residuales de la materia incandescente que constituyó las etapas iniciales y la contribución calórica proveniente de la desintegración de elementos radiactivos de vida prolongada [17].

### **1.2.3- Energía solar**

Nuestro planeta recibe del sol una cantidad de energía anual de aproximadamente 1,6 millones de kWh, de los cuales sólo un 40% es aprovechable, una cifra que representa varios cientos de veces la energía que se consume actualmente en forma mundial; es una fuente de energía descentralizada, limpia e inagotable. El aprovechamiento energético está entonces condicionado por la intensidad de radiación solar recibida por la tierra, los ciclos diarios y anuales a los que está sometida y las condiciones climatológicas del lugar.



Se define energía solar a aquella que mediante conversión a calor o electricidad se aprovecha de la radiación proveniente del sol; otra forma de aprovechamiento asociado considera la posibilidad de hacer uso de la iluminación natural y las condiciones climatológicas de cada emplazamiento en la construcción de edificios mediante lo que se denomina arquitectura bioclimática [16]

Hay varias maneras de recoger los rayos que cada día emite el Sol sobre la Tierra, estas formas determinan los distintos tipos de energía solar que mencionaremos a continuación:

- La energía solar fototérmica,
- La energía solar fotovoltaica
- La energía termoeléctrica.

#### **1.2.3.1- Energía solar fototérmica**

Este tipo de energía aprovecha el calor en sí mismo gracias a unos colectores solares que reciben el calor del sol y lo transfieren a un fluido de trabajo. El calor se utiliza entonces para calentar edificios, agua, mover turbinas, secar granos o destruir desechos peligrosos, entre otros usos.

#### **1.2.3.2- Energía solar fotovoltaica**

Este tipo de energía transforma los rayos en electricidad a través de los paneles solares o de las células fotovoltaicas. Los paneles solares, que están fabricados por silicio (el segundo elemento más abundante de que disponemos, después del oxígeno) que, junto con otros materiales, y al ser excitado por la luz solar, permite que se muevan los electrones y se genere una corriente eléctrica directa. Las células fotovoltaicas, por su parte, atrapan los fotones de la luz solar liberando con ello una carga que se convierte en electricidad [17].

#### **1.2.3.3- Energía solar termoeléctrica**

Este tipo de energía transforma el calor solar en energía eléctrica de una forma indirecta, ya que se trata de una combinación de las dos primeras: se aprovecha la energía solar fototérmica para obtener electricidad. Para ello, utilizan grandes sistemas de espejos móviles que concentran rayos solares en un punto específico calentando así un fluido. Ese fluido luego se aprovecha para producir electricidad mediante un generador [17].

#### **1.2.3.4- Energía solar - Conclusión**

Podría decirse que la energía solar es una de las más limpias de las que se utilizan hoy en día, puesto que no produce ruido ni contaminación, además tiene una larga (por no decir inagotable) vida. Sin embargo, también tiene alguna desventaja, y es que está sometida a constantes fluctuaciones: cuando más la necesitamos, es decir, en invierno, es cuando menos puede darnos, puesto que la radiación del Sol es menor. Asimismo, también depende, por supuesto, de la situación geográfica, por lo que no todos los países, ni siquiera todas las ciudades, pueden disfrutar de ella con igual accesibilidad. Además a pesar de los avances, la tecnología para su aprovechamiento todavía es cara para el consumidor [17].

#### **1.2.4- Energía de la biomasa**

Se entiende por tal al conjunto de materia orgánica renovable de origen vegetal, animal o procedente de la transformación natural o artificial de la misma. Como fuente de energía presenta una enorme versatilidad, permitiendo obtener mediante diferentes procedimientos tanto combustibles sólidos como líquidos o gaseosos (biogas, bioalcohol, etc.).

La biomasa de origen vegetal proviene de la reacción de la fotosíntesis, que sintetiza sustancias orgánicas a partir del CO<sub>2</sub> del aire y de otras sustancias simples, aprovechando la energía del sol. Por lo tanto desde el punto de vista ambiental, el aprovechamiento energético de la biomasa no contribuye al aumento de los gases de efecto invernadero, dado que el balance de emisiones de CO<sub>2</sub> a la atmósfera es neutro. En efecto, el CO<sub>2</sub> generado en la combustión de la biomasa es reabsorbido mediante la fotosíntesis en el crecimiento de las plantas necesarias para su producción y, por lo tanto, no aumenta la cantidad de CO<sub>2</sub> presente en la atmósfera [16].

El poder calorífico de la biomasa depende mucho del tipo de biomasa considerada y de su humedad. Así normalmente estos valores de poder calorífico de la biomasa se pueden dar en base seca o en base húmeda [16]. En general se puede considerar que el poder calorífico de la biomasa puede oscilar entre los 3000 – 3500 kcal/kg para los residuos ligno - celulósicos, los 2000 – 2500 kcal/kg para los residuos urbanos y finalmente los 10000 kcal/kg para los combustibles líquidos provenientes de cultivos energéticos.

Estas características, juntamente con el bajo contenido de azufre de la biomasa, la convierten en un producto especialmente atractivo para ser aprovechado energéticamente [16].

Los usos de la biomasa en aplicaciones energéticas son principalmente la producción de gas, energía calorífica (térmica) y energía eléctrica, actualmente la biomasa se está utilizando principalmente para producir electricidad e inyectarla a la red, mediante plantas de cogeneración eléctrica que aprovecha los residuos energéticos

#### **1.2.5- Energía de biocarburantes o biocombustibles**

Es una mezcla de hidrocarburos que se utiliza como combustible en los motores de combustión interna. Deriva de la biomasa, materia orgánica originada en un proceso biológico, espontáneo o provocado, utilizable como fuente de energía.

Para la obtención de los biocarburantes se pueden utilizar especies de uso agrícola tales como el maíz o la mandioca, ricas en carbohidratos o plantas oleaginosas como la soja, girasol y palmas. También se pueden emplear especies forestales como el eucalipto y los pinos.

En Europa, Argentina y Estados Unidos ha surgido diversa normativa que exige a los proveedores mezclar biocombustibles hasta un nivel determinado. Generalmente los biocombustibles se mezclan con otros combustibles en cantidades que varían del 5 al 10%. En nuestro país la Ley 26093 – Ley de biocombustibles, permite un 5% como mínimo.

#### **1.2.6- Energía marina**

O energía de los mares o mareomotriz (también denominada a veces energía de los océanos o energía oceánica) se refiere a la energía renovable producida por las olas del mar, las mareas, la salinidad y las diferencias de temperatura del océano. El movimiento del agua en los océanos del mundo crea un vasto almacén de energía cinética o energía en movimiento. Esta energía se puede aprovechar para generar electricidad que alimente las casas, el transporte y la industria.

### **1.2.7- Energía hidráulica**

Se obtiene del aprovechamiento de las energías cinética y potencial de la corriente de ríos o saltos de agua. Se trata de un tipo de energía sustentable cuando su impacto ambiental es mínimo y usa la fuerza hídrica sin represarla. Esto permite no afectar los ecosistemas de la zona.

Comparada con otras fuentes renovables, la hidroeléctrica se caracteriza por poseer mayor tradición tecnológica, factor de utilización y previsibilidad en la disponibilidad del recurso [16].

Constructivamente los aprovechamientos hidroeléctricos corresponden esencialmente a tres grandes esquemas:

- De embalse, con central a pie de presa.
- De pasada, y alta caída, que aprovechan la pendiente del terreno.
- De pasada, y baja caída, construidos sobre ríos de llanura y canales de riego

Los aprovechamientos de pasada no requieren reservorio, pero al carecer de capacidad de almacenamiento son altamente dependientes de las variaciones naturales del caudal del río. Contrariamente, los aprovechamientos a embalse pueden ser controlados y ajustarse rápidamente a las variaciones de demanda, que es una característica muy valiosa para el sistema eléctrico.

Es una fuente energética limpia y autóctona pero para la que se necesita construir infraestructuras necesarias que permitan aprovechar el potencial disponible con un coste nulo de combustible. El problema de este tipo de energía es que depende de las condiciones climatológicas.

### **1.3 - Redes Eléctricas Inteligentes (REI) o Smart Grids.**

Otra tecnología importante que se enfoca en la necesidad de administrar mejor los recursos energéticos y favorecer la protección del medio ambiente son las Redes Eléctricas Inteligentes o Smart Grids, estas constituyen una poderosa herramienta para afrontar los futuros desafíos que impone el sector energético actual.

Una Red Eléctrica Inteligente o Smart Grid es una red de transporte y distribución de energía eléctrica que tiene la capacidad de entender, asimilar, elaborar información y utilizarla adecuadamente, mediante el uso intensivo de las tecnologías de la información y las comunicaciones, se incorporaran a la red eléctrica tradicional dispositivos electrónicos tales como medidores, sensores o mandos, vinculados a través de distintas tecnologías de comunicación, logrando la centralización y uso de la información para provecho de todos los actores involucrados y así optimizar.

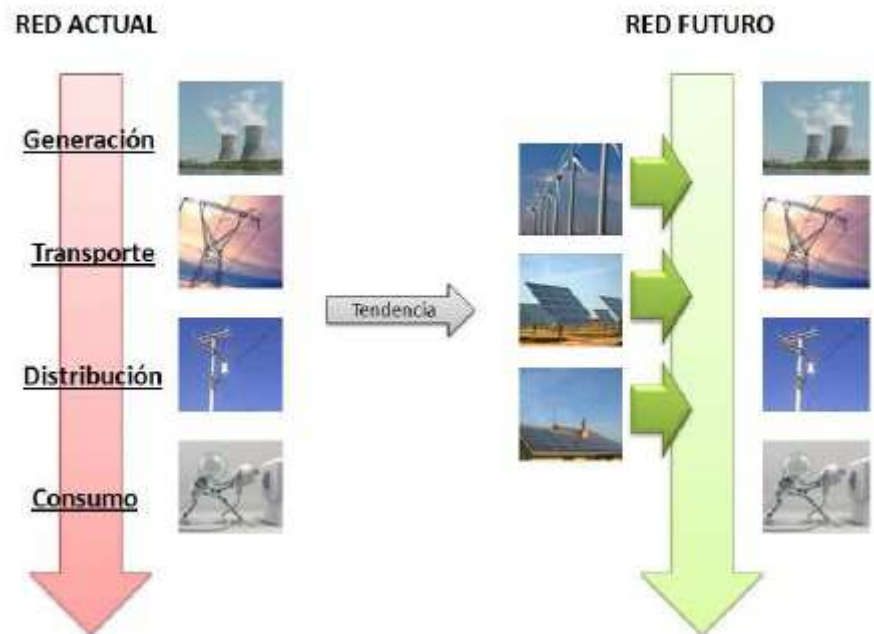
Este sistema permite la comunicación bidireccional entre el consumidor final (usuarios particulares o industriales) y las compañías eléctricas, de manera tal que la información proporcionada por los consumidores es utilizada por las compañías para permitir una operación más eficiente de la red eléctrica [9] [10].

Un aspecto importante asociado a las redes inteligentes es que deben facilitar la incorporación de las energías renovables, permitiendo diversificar el uso de las energías, considerando las propias limitaciones de las fuentes de energía alternativas, como su producción de forma irregular y la falta de infraestructura eléctrica en sus ubicaciones habituales [9].

Su objetivo es elevar la eficiencia, confiabilidad, calidad de servicio y producto. Disminuir el consumo de combustibles fósiles y la emisión de CO<sub>2</sub> [13].

Teniendo en cuenta estos aspectos entre otros, en el marco mundial la Comisión Europea se reunió en 2008 para elaborar el plan conocido como Plan 20-20-20. La estrategia 20-20-20 es una iniciativa lanzada para luchar contra el cambio climático con un objetivo claro: reducir las emisiones de gases invernadero en un 20%; ahorrar un 20% en el consumo energético; y proveer al sistema energético con al menos un 20% de renovables; siendo éste un propósito a lograr para el año 2020.

En la Figura N° 1.8 se presenta el nuevo modelo energético [11] [12].



**Figura N° 1.8 - Nuevos elementos en el modelo energético.**

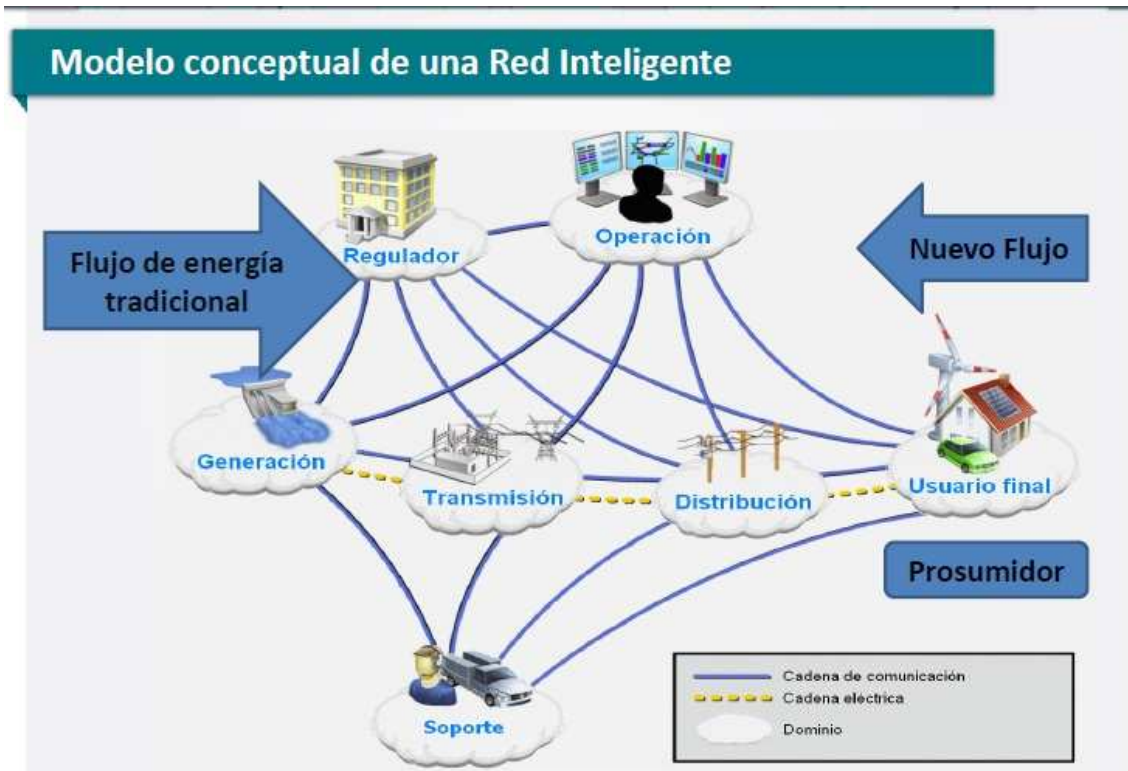
Todas las características de una Smart Grid la transforman en un sistema inteligente adquiriendo datos, comunicando, procesando información y ejerciendo control mediante una realimentación que le permite ajustarse a las variaciones que puedan surgir en un funcionamiento real, aplicando a la red toda esta funcionalidad, se podrán conseguir los fines propuesto para el mejoramiento del impacto ambiental que provoca la generación de energía en todo el mundo.

En nuestro país el uso de este tipo de redes, permitiría entre otros beneficios, la disminución de las importaciones y el uso más eficiente de todas las energías renovables.

Para llevar a cabo el desarrollo de la red eléctrica inteligente, será necesario realizar una fuerte inversión en investigación aplicada a la demanda y la generación, así como

en las tecnologías necesarias para realizar la implementación de la red de telecomunicaciones que sustente la transferencia de datos necesaria para la monitorización y control de la red. La cooperación entre las universidades y centros de investigación, compañías eléctricas, fabricantes, reguladores y legisladores será imprescindible para implementar el desarrollo tecnológico requerido.

En la Figura N°1.9 se presenta un modelo conceptual de una red inteligente:



**Figura N° 1.9 - Modelo conceptual de una Red Eléctrica Inteligente**

Nuestro país trabaja en varios proyectos pilotos en el marco de mejoramiento del uso eficiente de la energía para la protección socio-ambiental mediante el uso de energías renovables y las REI [9] [10].

Se pueden mencionar los siguientes proyectos:

- Proyecto en la Coop. de Amstrong, Prov.de Santa Fe. El proyecto se desarrollará en dos etapas: la primera consta de la instalación de los sistemas de telecontrol, tele supervisión y tele medición en 1000 puntos aproximadamente. En la segunda etapa se incorporan la generación distribuida (solar, mini-eólico, mini-hidro, biogás), la gestión activa de demanda y el alumbrado público eficiente
- Proyecto en la Coop. de Trenque Lauquen, Prov.de Bs.As con los mismos objetivos que el mencionado anteriormente, siendo esta una ciudad más grande que la anterior, con fuerte presencia de la agroindustria.

Se han formado grupos de trabajo entre Secretaría de Energía, Cammesa, ADEERA, e INTI para estudiar todas las cuestiones relacionadas con las Redes Eléctricas Inteligentes, su modo de aplicación y su relación con las distintas energías renovables, posibles de aplicación, en nuestro país [9] [10].

#### **1.4- Medidas a desarrollar para mejorar la eficiencia energética en la industria agropecuaria.**

Como se menciona en este proyecto en todo momento estamos frente a la obligación de un uso del consumo racional y eficiente de la energía en todos los sectores de producción. Hay un escenario energético mundial que obliga a poner en marcha iniciativas y políticas que moderen la demanda energética.

Nuestro proyecto pone interés en el sector agropecuario, dado que el consumo de energía sin los controles correspondiente, es uno de los factores que provoca efectos negativos sobre la competitividad (costes) de los productos y sobre el medio ambiente, a raíz de las emisiones de CO<sub>2</sub> producidas por los combustibles fósiles para su generación.

Por lo tanto, es necesario encontrar la forma de disminuir este consumo de combustibles fósiles sin frenar la producción de energía necesaria para el desarrollo económico de un país. Existen dos caminos para obtener dicho objetivo: producir la energía mediante fuentes renovables y reducir la demanda energética mediante el uso de equipos, técnicas y desarrollo más eficientes. Para ello es primordial conocer los parámetros que deben ser monitoreados, sus características y como mejorarlos.

##### **1.4.1-Industria avícola**

En el sector avícola, para contribuir a la mejora de la eficiencia energética, deberán considerarse una serie de factores directamente implicados en las características de los lugares de producción, entre ellos podemos mencionar:

- Las condiciones climáticas de la región.
- Las particularidades locales de la zona.
- Las necesidades de ambiente en el interior de los alojamientos según especie.
- Estanqueidad de las naves
- La relación coste-eficacia de las inversiones en los elementos y máquinas que conforman la instalación avícola.

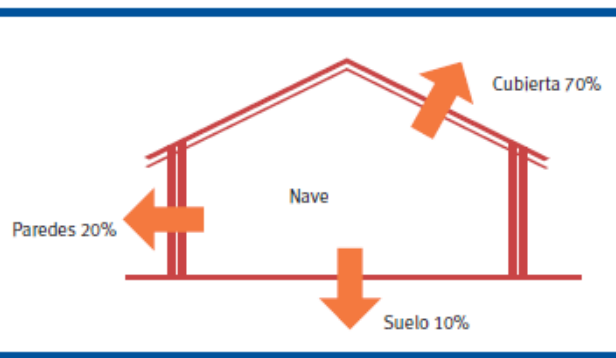
Las recomendaciones para dicha mejora estarán basadas en una combinación de normas de construcción y equipamiento de las granjas: materiales de construcción, aislamientos, ventilación, calefacción e iluminación principalmente, de tal forma que se pueda garantizar un funcionamiento integrado de los diferentes sistemas sin que el funcionamiento de alguno de ellos suponga un derroche de energía [18].

Los aspectos básicos sobre los que se incide y que se consideran más importantes para contribuir al ahorro y eficiencia energética, son los siguientes:

- Aislamiento adecuado de los edificios.
- Regulación correcta de los equipos de climatización de las naves.
- Iluminación eficiente.
- Revisión y mantenimiento de los equipos.
- Implantación de barreras vegetales cortavientos.

Considerando **el aislamiento** en el sector avícola, es necesario tener una refrigeración adecuada al momento de la crianza, los establecimientos deben poseer buenas tecnologías de aislamiento, para lograr un buen ahorro energético. Esto permitirá reducir las pérdidas de calor en tiempo de frío y reducir las ganancias de calor en épocas calurosas [18], la Figura 1.10 nos ejemplifica las pérdidas típicas.

Mejorando la calidad del ambiente interior, los sistemas de climatización funcionarán a un régimen moderado, reduciendo el gasto energético.



**Figura Nº1.3 - Pérdidas de energía que se producen en las cubiertas.**

Es importante recordar, que el equilibrio térmico en el alojamiento depende también de otros factores como son el calor suministrado por los propios animales y la necesidad de calentar o enfriar el aire que entra a través de los sistemas de ventilación.

Por lo tanto, **la ventilación** es otro factor fundamental en la avicultura. La ventilación tiene por objeto: aportar oxígeno al interior de la nave, eliminar CO<sub>2</sub>, polvo y otros gases, eliminar vapor de agua producido por la respiración y las heces y controlar la temperatura del alojamiento.

Existen diferentes sistemas. Podemos mencionar:

- La **ventilación natural** que consiste en el manejo de ventanas manuales o automáticas para renovar el aire del interior de la nave aprovechando la velocidad y presión que el aire ejerce sobre la nave. Su principal ventaja es el ahorro de energía y la baja inversión en equipos, sin embargo son difíciles de gestionar.
- La **ventilación dinámica**: se realiza con la ayuda de ventiladores que mueven el aire necesario en cada fase de la producción. Con este tipo de ventilación se consigue una buena gestión del ambiente e independiente de la climatología, pero requiere una mayor inversión inicial y un mayor consumo energético.

Para control de la ventilación dinámica se pueden usar temporizadores en los ventiladores para que funcionen por tiempos programados, pero no tienen en cuenta la temperatura de la nave. Son baratos, inexactos, inseguros y poco efectivos. También se usan termostatos que controlan únicamente la temperatura, sin tener en cuenta otros parámetros, accionando los ventiladores todo (100%) o nada (0%), que requieren mucho control por parte del avicultor.

Para lograr un buen control de la ventilación dinámica en las diferentes salas de una explotación y atendiendo a las necesidades de los animales a lo largo de su crecimiento, es conveniente la instalación de buenas cajas electrónicas de programación, o comúnmente llamadas "reguladores", y conocer su funcionamiento. Estos se deben ajustar al mínimo caudal necesario, se deben realizar inspecciones y limpiezas periódicas de los equipos para disminuir su consumo. Estas limpiezas son recomendables realizarlas en forma semanal debido al importante nivel de suciedad de las granjas avícolas.

En la actualidad este control, así como tantos otros, pueden ser realizados por un sistema informático, siendo el que aporta mayores beneficios.

**La iluminación** es otro de los parámetros que se debe de tener en cuenta para el uso eficiente y ahorro de la energía. En zonas donde no se llevan a cabo actividades que exijan una iluminación de gran calidad se debe usar un nivel de iluminación medio.

Por lo tanto el encendido de un menor número de lámparas (o de lámparas de menor potencia) conllevará un ahorro energético.

Es recomendable por tanto la instalación de iluminación de alta eficiencia energética y bajo consumo. Lo cual supone desechar si es posible las lámparas incandescentes y utilizar iluminación con tubos fluorescentes o lámparas fluorescentes electrónicas de bajo consumo. Por término medio las lámparas fluorescentes compactas (LFC) consumen un 80% menos de electricidad que las incandescentes, duran hasta 12 veces más y reducen los costes de mantenimiento ya que necesitan ser cambiadas con menor frecuencia [18].

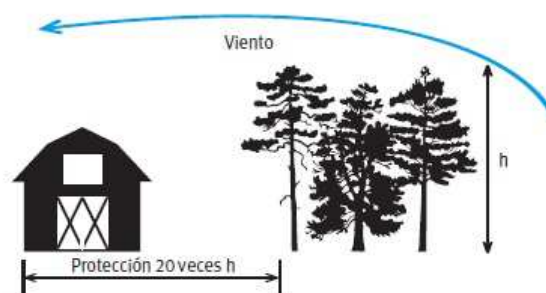
En zonas de acceso puntual tales como accesos a las naves, es conveniente instalar sensores de presencia para asegurar que las luces de estas zonas no permanecen encendidas más tiempo del necesario [18].

Debe ser frecuente la limpieza de la suciedad acumulada en lámparas y difusores, ya que esta disminuye de manera importante la luz emitida, lo cual puede provocar que se enciendan o se instalen más puntos de luz de los necesarios.

También se debe considerar la utilización de pinturas blancas o de tonos claros, y mantenerlas siempre limpias.

Un sistema de iluminación bien diseñado permitirá reducir el consumo de electricidad, manteniendo el nivel de iluminación necesario e incluso mejorarlo.

Otra forma que ayuda a la mejora del uso racional de la energía es la implementación de barreras cortavientos naturales, las cuales pueden existir de antemano o bien pueden ser creadas, tal como se ve en la Figura N°1.11.



**Figura N°1.11 - Protección mediante barreras vegetales cortavientos.**

Estas barreras facilitan la gestión de la ventilación en naves que poseen ventilación natural, además dichas naves se encuentran menos expuestas a los vientos, reduciendo así las pérdidas energéticas por ventilación y por los posibles ocasionados por la propia acción del viento sobre las superficies exteriores.

Calcular una protección cortavientos es una tarea compleja, pues depende de muchos factores y es una práctica muy imprecisa.



Para la protección de naves con sistemas de ventilación natural, la recomendación es la siguiente: una plantación vegetal donde la permeabilidad al aire estimada sea del 50%, ofrece una protección a los vientos en una distancia aproximadamente igual a 20 veces su altura como se aprecia en la Figura N°1.11 [18].

#### 1.4.2- Industria ganadera:

Mirando el sector ganadero en general, las consideraciones que se deben tener en cuenta para el ahorro y eficiencia energética son similares a los ítems analizados para el sector avícola, pero dependerán en cada caso de los parámetros específicos a monitorear y controlar en cada uno de ellos.

En el aislamiento se deben considerar las normas establecidas para cada tipo de especie ganadera y en función de las diferentes edades o estados fisiológicos. No se debe ahorrar en la calidad de los materiales de aislamiento, ya que ello beneficia al mejor rendimiento de los equipos y el consecuente ahorro energético. Cuidar sobre todo, el aislamiento de la cubierta, ya que por allí se produce el 70% las pérdidas de energía. Un mejor aislamiento permite no sobredimensionar la ventilación y la refrigeración, esto aporta una mejor relación coste-beneficio.

En la climatización se debe tener un buen sistema de regulación que atienda las necesidades ambientales de los animales en cada estadio. Los reguladores deben estar ajustados a los caudales mínimos necesarios. Se debe implementar a las naves con sistemas automáticos de regulación, reguladores y sistemas informáticos, siempre que fuese posible. Los sistemas de calefacción tipo pantallas, tiene un menor consumo y son sistemas de calor más eficientes [18].

En iluminación, sustituir lámparas incandescentes por fluorescentes. Donde se requiera mayor iluminación instalar lámparas de bajo consumo (fluorescentes compactas). Utilizar superficies blancas o tonos claros y mantenerlas limpias.

En la Figura N°1.12 se da una comparación de los distintos tipos de lámparas, en cuanto al consumo energético.

Tipo de Lámpara	Flujo Luminoso (lum)	Potencia (W)	Eficacia (lum/W)
 Bombilla incandescente	715	60	11,9
	1.350	100	13,5
 Tubo fluorescente	1.150	18	63,9
	2.850	36	79,1
	4.600	58	79,3
 Lámpara fluorescente	575	11	52,0
	855	15	57,0
	1.140	20	57,0
	1.450	23	63,0

**Figura N°1.12– Tipos de lámparas**

En cuanto a la estanqueidad, cuidar las entradas de aire parásitas, ya que aumenta el consumo de la calefacción o refrigeración calculada para las naves, provocando un mal consumo de energía[18].

Un factor importante en todos los casos de las instalaciones ganaderas, es la organización de un programa de revisión y mantenimiento preventivo de todos los

equipos que son utilizados en la explotación y el cumplimiento de dicha tarea, como una obligación más de la producción.

#### **1.4.3- Industria agrícola:**

En agricultura se están desarrollando nuevos métodos de cultivos controlados para llamados agricultura de precisión (AP) que permiten ayudar al agricultor a conseguir una mayor eficiencia productiva, a la vez que a reducir el impacto medioambiental. La creación de mapas del punto de funcionamiento del motor (régimen de giro, par) durante su trabajo en parcela permite optimizar el trabajo, disminuyendo también el consumo de combustible. Los recientes sensores que estiman las características del suelo proporcionan herramientas para controlar con precisión y monitorear sobre la marcha las variaciones de parámetros relacionados con la fertilidad. El hecho de poder registrar con gran exactitud los tiempos de trabajo, distancias, consumos (en gasóleo y otros insumos), áreas trabajadas, parámetros del motor, máquinas empleadas, etc. gracias a los sensores y monitores instalados en las maquinarias utilizadas, supone una herramienta poderosísima para la gestión de una explotación agraria, permite en detalle el uso de recursos y personas. El ahorro que se obtendrá es difícil de estimar en términos generales, pero la información obtenida puede influir en decisiones tan importantes como la adquisición o no de un tractor adicional con el consecuente ahorro de combustible necesario para determinada explotación [18].

El uso de los equipos combinados de siembra, como el que se puede ver en la FiguraNº1.13, permite reducir el tiempo dedicado al laboreo en las grandes explotaciones dotadas además de tractores más potentes, y por otra, en un intento de evitar el efecto negativo de la lluvia entre la preparación de la tierra y la siembra (con los equipos combinados, “parcela preparada, parcela sembrada”). Con el resultado añadido de un ahorro energético y económico [19].



**Figura Nº1.6–Equipo combinado de siembra.**

En lo que respecta a los invernaderos, las consideraciones a tener en cuenta para el ahorro energético en este sector, son similares a las del sector ganadero.

Se debe tener en cuenta las características constructivas referidas a la orientación, siendo conveniente la Este-Oeste para maximizar la captación solar, la altura mínima para aumentar la inercia térmica y facilitar la ventilación, la anchura máxima que contribuye a mejorar la ventilación, la separación mínima entre invernaderos, recomendada en 3 metros, para facilitar el flujo de aire para la ventilación natural y las tuberías de calor enterradas, para disminuir las pérdidas de calor fuera del invernadero [20].

Para la estanqueidad asegurar el aislamiento de la cubierta y el correcto cierre de ventanas, utilizar pantallas térmicas en las paredes laterales, disponer de mallas cortavientos que se puedan retirar en los periodos cálidos para no disminuir la ventilación.

Utilizar buenos materiales de aislamiento en las cubiertas, que permitan una buena transmisividad a la radiación solar, superior al 80% y una transmisividad máxima a la radiación infrarroja inferior al 50%. Blanquear el invernadero en periodos calidos para disminuir las necesidades de refrigeración.

Mantener una buena ventilación con una correcta ubicación de las ventanas teniendo en cuenta los vientos dominantes en periodos calidos. La superficie de ventilación debe ser de al menos un 30% de la superficie del suelo cubierto por el invernadero y la porosidad de las mallas anti-insecto debe ser superior al 40% para no disminuir la ventilación natural.

En iluminación artificial se deben reemplazar las lámparas incandescentes por las fluorescentes.

### **1.5-Soluciones tecnológicas para mejorar la eficiencia energética**

Tal como se ha detallado en el apartado anterior, la eficiencia energética en establecimientos agropecuarios, como ser avícolas, ganaderos e invernaderos, se puede obtener mediante:

- Producción la energía mediante fuentes renovables.
- Reducción la demanda energética a través del uso de equipos, técnicas y desarrollo más eficientes.

Ejemplos del uso de energía eólica se pueden ver en el proceso de electrificación del espacio rural del Partido de Tandil, proceso llevado adelante por la Cooperativa Rural Eléctrica de Tandil-Azul Limitada (CRETAL) [16], así como ejemplos de energía fotovoltaica para la agricultura y el desarrollo sostenible se pueden ver en el trabajo realizado por B. van Campen, D. Guidi y G. Best sobre la “Energía solar fotovoltaica para la agricultura y desarrollo rural” [26], Figura N°1.14, mientras que los biocombustibles pueden ser utilizados en las calderas que se usan para calefacción de los establecimientos.



**Figura N°1.14–Equipo combinado de siembra**

Los biocombustibles más utilizados para el uso de biomasa con aplicaciones para calefacción son los pellets, las astillas de madera y los residuos agroindustriales como los orujillos, las pepitas de uva o las cáscaras de almendra [18].

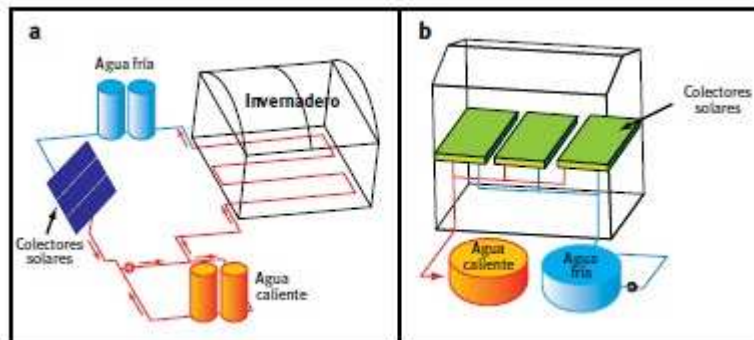
Teniendo en cuenta las necesidades energéticas del establecimiento se puede plantear el uso de energía solar térmica, como apoyo a la calefacción y energía solar fotovoltaica para producir electricidad.

En las Figura 1.15 se puede ver un esquema de la aplicación de un sistema solar térmico:



**Figura N°1.15 – Sistemas de calefacción**

Mientras que la Figura 1.16 muestra esquemas de sistemas de calefacción mediante colectores solares (a) y un sistema de absorción (b):



**Figura N°1.16– Sistemas de calefacción [24].**

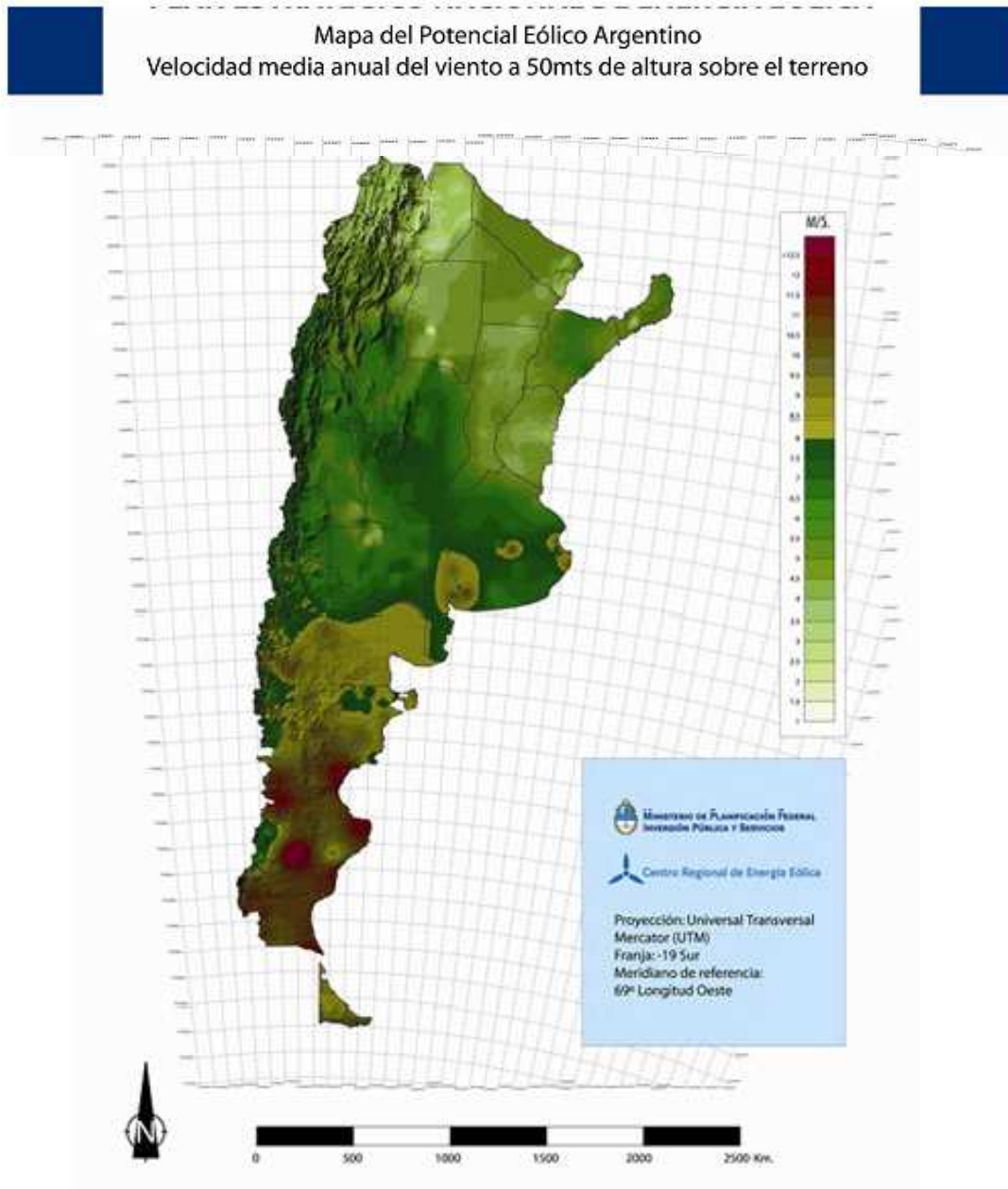
Para la implementación de estos sistemas se deben considerar:

- Condiciones climáticas.
- Condiciones constructivas de los establecimientos
- Tecnología utilizada en los establecimientos

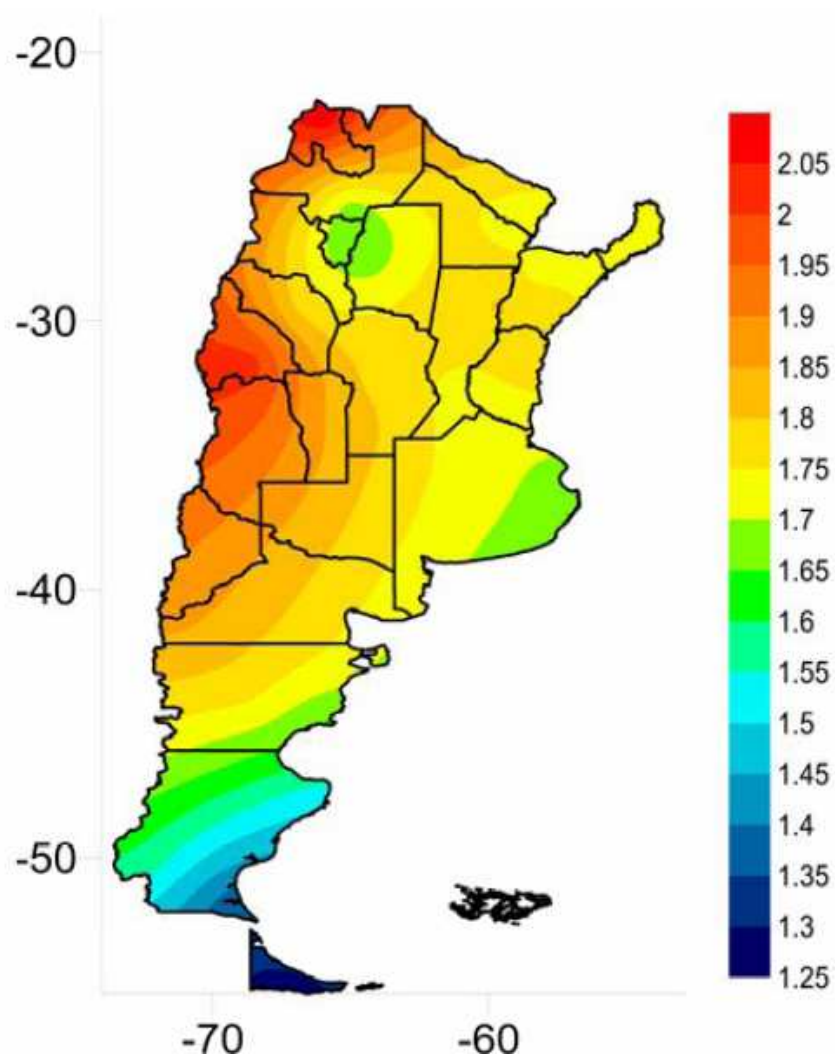
Las condiciones climáticas influirán tanto en la construcción de los establecimientos, así como también en la selección de la fuente renovable para la generación de energía. No será lo mismo la un sistema eólico en la Pampa o en Entre Rios, donde

podemos encontrar velocidades de viento que oscilan entre 4 y 8 m/s, que en Santa Cruz donde se encuentran velocidades del orden de 9 m/s e inclusive lugares con velocidades mayores a 12,5 m/s, esto se ve claramente en la Figura N°1.17 donde se presenta un mapa con las velocidades del viento en la Argentina [22].

De forma similar no será lo mismo un sistema fotovoltaico en San Juan que en Buenos Aires, esto se puede ver en la Figura N 1.18, donde los valores presentados son en MWh/m<sup>2</sup> [23]. Así como también en los procesos de biodigestión, donde la temperatura ambiente tiene una influencia muy importante.



**Figura N° 1.17- Mapa del Potencial eólico Argentino**



**Figura N° 1.18- Mapa del Potencial fotovoltaico Argentino.**

Una vez seleccionada la fuente de energía renovable más adecuada a la zona y al tipo de establecimiento agropecuario, se deben pensar en las distintas tecnologías que mejoren la producción y el aprovechamiento de la energía generada.

Como ejemplo de esto podemos nombrar, en el caso de sistemas fotovoltaicos, sistemas de seguimiento solar, los cuales permiten lograr hasta un 25 [38] más de energía generada. Sistemas de carga de batería con seguimiento de punto máximo de potencia, los cuales permitirían aprovechar al máximo la energía generada [39]. En el caso de sistemas generación eólica, la medición de la velocidad del viento es importante ya que permite la valoración de un sitio para generación de energía eléctrica, como para el testeado de aerogeneradores de eje horizontal para certificación de su curva de potencia [40]. Así como también sistemas de control de temperatura, presión y PH en sistemas de biodigestiones, los que permitirán una mayor y más rápida generación de energía, un ejemplo de esta la producción de gas metano para el uso en sistemas de calefacción [41].

Por otro lado, toda fuente de energía renovable debe contar con un sistema de almacenamiento de energía, el cual le permita almacenar la energía generada y disponer de ella cuando sea necesaria. En función del dimensionamiento de dicho sistema se podrá contar, por un determinado tiempo, con energía aun cuando las

condiciones ambientales no sean las favorables para la generación. Las baterías es el medio de almacenamiento de energía más común, las cuales entregaran la misma en forma de una corriente continua.

De esto surge que otro sistema importante en la generación de energía a través de fuentes renovables es el inversor (inverter). Sistema el cual convierte una corriente eléctrica continua en una corriente eléctrica alterna, monofásica o trifásica, para alimentar los dispositivos encargados de los procesos.

Pero como las condiciones ambientales están fuera de nuestro alcance es importante contar con un sistema de respaldo, pudiendo ser este un grupo electrógeno el cual debería contar con un sistema de transferencia automática para evitar inconvenientes en los distintos procesos.

Y finalmente el concepto de Internet de las cosas, donde un sistema informático y un conjunto de sensores y actuadores permitirá manejar en forma completa el establecimiento, monitoreando los distintos parámetros climáticos, como ser la temperatura y la humedad relativa en diferentes puntos del interior del establecimiento, la temperatura exterior y la velocidad del aire en el interior y exterior, y en función de estos parámetros actuar, evitando consumo de energía innecesaria. Estos sistemas pueden permitir, además, el uso alternativo de diferentes tipos de energías combinando las de origen fósil, no renovable, con las alternativas distintas energías renovables eólica, biomasa, solar fotovoltaica.

#### **1.6- Alcance del proyecto**

Utilizando el concepto de Internet de las cosas se pretende lograr un sistema inteligente para la administración de la energía en establecimientos agrícolas. Para esto se estudiará algunos de los sistemas de generación de energía renovable (fotovoltaicos, eólicos y de biodigestión). También los dispositivos electrónicos que ayuden a mejorar la generación de energía, su almacenamiento y posterior reutilización (sistemas de seguimientos, reguladores de carga, sistemas de control de temperatura, inversores y conmutadores).

## 2- Microestaciones

En el siguiente apartado, se detallan cuestiones vinculadas con el ensamble de las placas de Circuito Impreso correspondientes a las Microestaciones Meteorológicas y las pruebas realizadas sobre las distintas etapas las mismas.

### 2.1- Etapa de Fuente de Alimentación

#### 2.1.1- Montaje

Una vez recibidos los Circuitos Impresos de las Microestaciones, procedimos a realizar el montaje y prueba de la etapa correspondiente fuente de alimentación de las mismas. Para ello se requirió de los componentes mencionados en la Tabla N° 2.1.

Referencia	Encapsulado	Cantidad	Descripción
<b>C7</b>	SM0805	1	Capacitor Cerámico 0,01 uF
<b>C6, C4, C12, C22</b>	SM0805	4	Capacitor Cerámico 0,1 uF
<b>C10</b>	SM0805	1	Capacitor Cerámico 0,22 uF
<b>C8</b>	SM0805	1	Capacitor Cerámico de 0,47 uF
<b>C13</b>	C_G2_D4	1	Capacitor Electrolítico de 1 uF x 16 V
<b>C1, C3, C5, C9, C11</b>	C_G2_D4	5	Capacitor Electrolítico de 10 uF x 16 V
<b>R12</b>	SM0805	1	Resistencia de 0Ω
<b>R3, R4, R5, R8, R10</b>	SM0805	5	Resistencia de 1 kΩ
<b>R22</b>	SM0805	1	Resistencia de 4,7 kΩ
<b>R2, R6, R7, R11</b>	SM0805	4	Resistencia de 10 kΩ
<b>R9</b>	SM0805	1	Resistencia de 22 kΩ
<b>R1</b>	SM0805	1	Resistencia de 33 kΩ
<b>RV2</b>		1	Preset Multivuelta de 20 kΩ
<b>RV1</b>		1	Preset Multivuelta de 47 kΩ
<b>P1,P2</b>		2	Bornera de 2 Contactos, P/Circuito Impreso
<b>K1</b>		1	Bornera de 3 Contactos, P/Circuito Impreso
<b>D1, D2</b>	DO-213AB	2	Diodo Rectificador 1N6484
<b>D4</b>	SM0805	1	LED Rojo Estándar
<b>D3, D5</b>	SM0805	2	LED Verde Estándar
<b>U1</b>	TO220_VERTICAL	1	Regulador de Tensión LM7805
<b>U2</b>	R-PQFP-N20	1	BQ24071
<b>U3</b>	TSOP-5	1	NCP700B - 200 mA BICMOS LDO Regulator

Tabla N°2.1

A continuación se mencionan los principales componentes que integran la Fuente de Alimentación:

- **BQ24071:** Circuito Integrado Cargador de Baterías de Litio;
- **NCP700B - 200 mA BICMOS LDO Regulator:** Regulador de tensión. Tensión de Salida = 3,3 Volts.



## 2.1.2-Detalle del Proceso de Prueba

Una vez realizado el soldado de los componentes sobre un total de seis placas de Circuito Impreso se procedió a alimentarlas mediante el uso de la Fuente de Alimentación de Laboratorio.

El proceso de prueba se inició conectando la Fuente de Alimentación de Laboratorio sobre los bornes del Conector P2 de una de las Placas de Circuito Impreso en ensayo.

Durante el proceso de ensayo se procedió a medir la tensión de salida del Regulador de Tensión NCP700B, encargado de entregar una tensión regulada a su salida de 3,3 Volts.

Luego de alimentado el circuito, se procedió a elevar la tensión entregada por la Fuente de Laboratorio, midiendo simultáneamente la Tensión entregada por el Regulador NCP700B. De allí se pudo observar lo siguiente:

- 1) Mientras que la Tensión entregada por la Fuente de Laboratorio se mantuvo por debajo de los 6 Volts, la tensión de salida del Regulador NCP700B se encontró alrededor de los 3,3 Volts, entregando valores normales de funcionamiento;
- 2) Una vez superados los 6 Volts de tensión de entrada, se pudo notar que la tensión entregada por el Regulador NCP700B no arrojaba valores dentro de los parámetros normales de funcionamiento, entregando valores mayores a 3,3 Volts, lo cual comenzó a denotar una anomalía en el funcionamiento del circuito.

Una vez estudiada la situación anterior, se pudo verificar que la causa de la anomalía se debió a la adquisición incorrecta del Circuito Integrado destinado a la Carga de la Batería de Litio, ya que, el componente soldado entrega a su salida una tensión nominal de 6 volts. Dicho valor se caracteriza por ser igual al valor máximo de tensión admisible a la entrada del Regulador NCP700B lo cual generó el mal funcionamiento de este. De lo anterior, podremos ver que se incurrió en el error de haber adquirido el componente electrónico incorrecto, ya que, en vez de haber montado el BQ24071, se debió haber montado el componente BQ24070.

## 2.2- Etapa de Sensores y Transceptor

### 2.2.1- Montaje

El siguiente paso en el armado y testeo de las placas de Circuito Impreso consistió en el montaje de los sensores y demás componentes destinados a la medición de los parámetros meteorológicos. Para ello se requirió de los componentes mencionados en la Tabla N° 2.2:

Referencia	Encapsulado	Cantidad	Descripción
<b>C20, C18, C17, C19, C24, C25, C14</b>	SM0805	7	Capacitor Cerámico 0,1 uF
<b>C15</b>	C_G2_D4	1	Capacitor Electrolítico de 1 uF x 16 V
<b>C16</b>	C_G2_D4	1	Capacitor Electrolítico de 4,7 uF x 16 V
<b>R13, R14, R16, R17</b>	SM0805	4	Resistencia de 10 kΩ
<b>P4</b>		1	Conector Macho a Cable Plano P/CI de 20 Vías

<b>U4</b>	LGA8	1	MPL3115A2 - Barómetro
<b>U5</b>	QFN24	1	SI7005 - Sensor de Humedad y Temperatura
<b>U6</b>	SOIC8	1	AT30TS75 - Sensor de Temperatura
<b>U7</b>	WSOF5	1	BH1721FVC-Ambient Light Sensor

**Tabla N° 2.2**

Luego de realizar el montaje de los componentes presentes en la Tabla N° 2.2 se realizó el montaje de los componentes vinculados con el transceptor, recurriendo a los módulos ATZB-24-B0 (2,45 GHz) los cuales se montaron sobre dos de las placas de Circuito Impreso (Tabla N° 2.3). La Figura 2.1 nos muestra el circuito totalmente montado.

Referencia	Encapsulado	Cantidad	Descripción
<b>C_RF1, C_RF2, C_RF3</b>	SM0402	1	Capacitor Cerámico de 22 pF
<b>R_RF2</b>	SM0805	5	Resistencia de 0Ω
<b>U8</b>		1	ATZB-24-B0 (2,45 GHz)
<b>U9</b>	Balun1	1	2450FB15L0001 (2,45 GHz)

**Tabla N° 2.3**

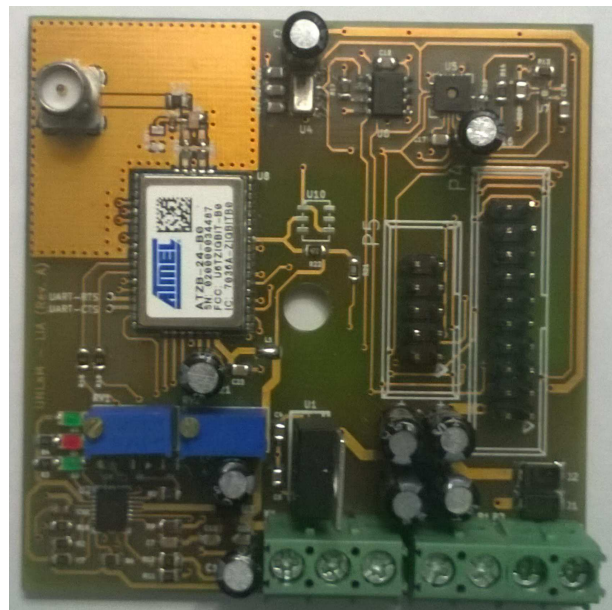


Figura 2.1. Circuito con ATZB-24-B0 montado.

Por otro lado, también se realizó el montaje de la microestación utilizando la solución en 900 MHz. Para ello se recurrió al uso de los componentes mencionados en la tabla N°2.4. La Figura 2.2 nos muestra el circuito totalmente montado.

Referencia	Encapsulado	Cantidad	Descripción
<b>C_RF1, C_RF2</b>	SM0805	1	Capacitor Cerámico de 68 pF
<b>R_RF1</b>	SM0805	1	Resistencia de 0Ω
<b>C_ANT1</b>	SM0204	1	Capacitor Cerámico de 100 pF
<b>U8</b>		1	ZIGBIT ATZB-900-B0
<b>U9</b>	Balun1	1	Balun 0896FB15A0100 (863-928 MHz)

Tabla Nº 2.4

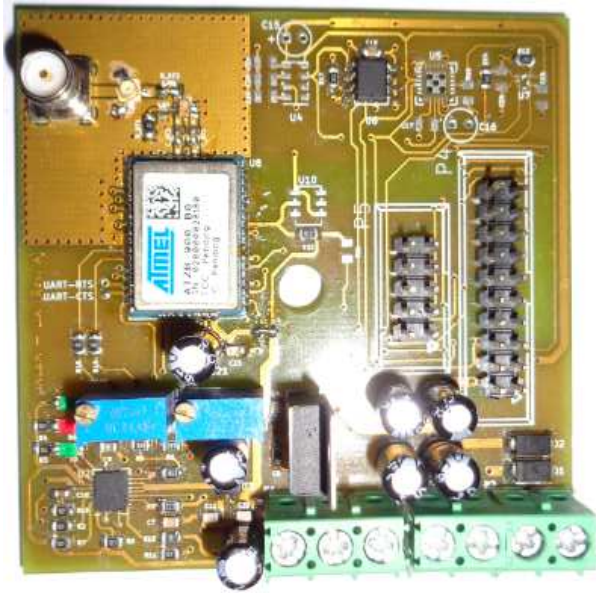


Figura 2.2. Circuito con ATZB-900-B0 montado.

### 2.2.2- Proceso de Prueba de los Sensores

Luego de realizar el montaje de los componentes anteriores sobre las placas de Circuito Impreso se procedió al ensayo de los sensores para lo cual se accedió a las señales del bus I2C mediante el uso de conector P4.

El proceso de prueba de los sensores se inició conectando la Fuente de Alimentación de Laboratorio sobre los bornes del Conector P2 de una de las Placas de Circuito Impreso en ensayo.

A continuación, mediante el uso de un Osciloscopio Digital, procedimos a medir la diferencia de potencial existente en las señales del bus I2C y GND. En condiciones normales de funcionamiento deberíamos haber medido una tensión de 3,3 Volts entre GND y CLK – SDA, lo cual no sucedió de tal forma, ya que en la totalidad de las placas ambas señales arrojaron valores de tensión por debajo de los 2 Volts. Luego de realizar una inspección visual se pudo observar que el Barómetro no se había montado en la posición correcta, por lo cual se procedió a remover dicho componente de la totalidad de las placas. Una vez realizada la tarea anterior se procedió a realizar nuevamente las mediciones sobre el bus I2C, de lo cual se obtuvo el siguiente resultado: en aquellas placas donde se había montado el Transceptor se observó que la falla persistía, mientras que en el resto de las placas, en donde solo se desmontaron los sensores, el bus I2C arrojó valores de tensión normales de alrededor de los 3,3 Volts. De lo anterior pudimos deducir que el origen de la falla, no solo se debió al montaje erróneo de los Barómetros, lo que es más, al montar en transceptor en dos de las placas, introdujimos un componente electrónico en el cual los pines destinados al control del bus I2C no se hallaban debidamente configurados para tal fin, originando así la falla sobre el bus de comunicaciones de los sensores.

De la inspección visual realizada sobre la etapa correspondiente a los sensores, no solo se pudo detectar el montaje erróneo de los Barómetros, contrastando con el esquemático del circuito se pudo observar la ausencia de la pista la cual debía conectar la señal VDDIO del Barómetro con la alimentación VCC. Dicho error se

corrigió realizando el trazado de la pista faltante sobre el archivo correspondiente al PCB de las placas de Circuito Impreso, dando origen a la Revisión. Por otro lado, en las seis placas existentes se cableó la pista faltante de forma manual mediante el uso de Alambre wire wrap.

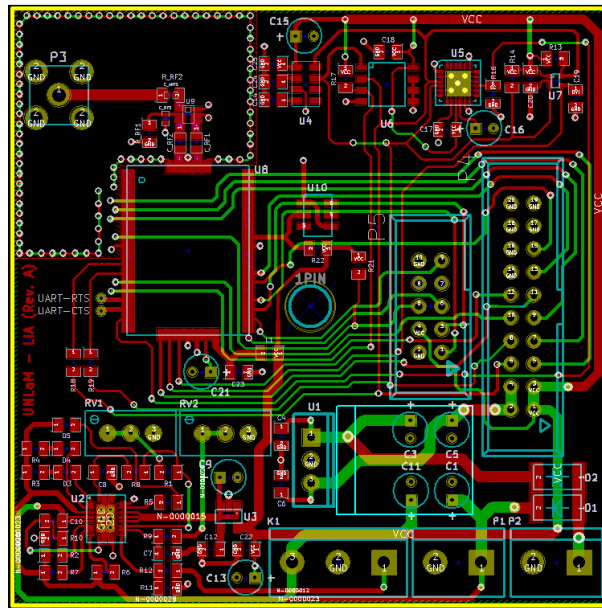


Imagen Nº 2.1- Placa de Circuito Impreso Revisión A

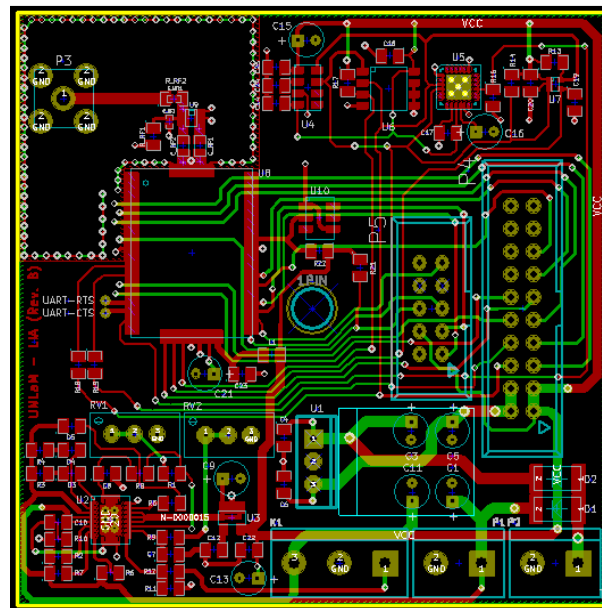


Imagen Nº 2.2- Placa de Circuito Impreso Revisión B

Una vez solucionados los inconvenientes anteriormente detallados, estuvimos en condiciones de adentrarnos en la programación de los sensores.

### 2.3- Lectura de los sensores

A continuación se manifiesta las características relevantes de cada sensor incluidos en la "Placa Microestación", necesarias para realizar la comunicación por el I2C Bus entre

el módulo ZigBit™ ATZB-24-A2/B0 de Atmel, configurado como Master y los sensores ambientales configurados como Slave.

Las funciones implementadas en el presente pueden no corresponderse por completo con las librerías ofrecidas por Atmel, debido a que las pruebas fueron realizadas con un maestro externo implementado en un PIC16F628 de Microchip. No obstante esto, por su similitud son una buena referencia al momento de ser implementadas en cualquier otra plataforma.

### 2.3.1- Resumen de sensores utilizados en la Microestación

DISPOSITIVO	SI7005	AT30TSE752	BH1721FVC	MPL3115A2
<b>Magnitud leída</b>	Humedad relativa Temperatura	Temperatura	Luz ambiente	Presión atmosférica Temperatura
<b>Dirección binaria</b>	1000000X	1001A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> X	0100011X	1100000X
<b>Dir. escritura</b>	0x40	0x90*	0x46	0xC0
<b>Dir. lectura</b>	0x41	0x91*	0x47	0xC1
<b>Mnemónico asignado</b>	SLV_ADD_HUM	SLV_ADD_TEMP	SLV_ADD_LUZ	SLV_ADD_PRES
<b>Rango de medición</b>	0 -100% RH	-55 a +125°C	1 a 65528Lux	50 a 110kPa
<b>Marca</b>	SiliconLabs	Atmel	ROHM Sem.	Freescale

**Tabla 2.6- Resumen de características**

\*En la placa Microestación A<sub>2</sub>, A<sub>1</sub> y A<sub>0</sub> fueron conectadas a masa, es decir 0 lógico.

### 2.3.2- Aspectos Preliminares

Para efectuar la puesta en marcha y prueba de los sensores se recurrió al uso de un microcontrolador PIC 16F1824 dada la facilidad y ductilidad en el uso de la herramienta de programación y compilación utilizada. Para ello se recurrió al compilador MPLAB X el cual permite incorporar de forma rápida las rutinas necesarias para implementar la comunicación entre los sensores y el microcontrolador. Dicha implementación se consiguió utilizando la herramienta MPLAB Code Configurator, logrando así implementar las rutinas de lectura y escritura de la interfaz I2C del microcontrolador PIC 16F1824. Una vez implementadas las rutinas mencionadas anteriormente, nos avocamos en el estudio de los registros asociados a la configuración y lectura de cada uno de los sensores. A continuación se describe en detalle cada uno de los sensores y los resultados obtenidos luego de su puesta en funcionamiento.

## 2.3.3- Sensor de temperatura AT30TSE75

### 2.3.3.1- Dirección I2C del Sensor

Con la finalidad de establecer la comunicación entre el Microcontrolador y el Sensor mediante sus interfaces I2C, el sensor de Temperatura posee la siguiente dirección: 0b10010000, 0x48 (Hexadecimal). Dicha dirección de memoria deberá ser antepuesta a todo mensaje transmitido desde el Microcontrolador hacia el Sensor en cuestión.

### 2.3.3.2-Descripción del Registro de Control

Para efectuar la puesta en marcha del Sensor de Temperatura, inicialmente se realizó la configuración del registro de control asociado al sensor. Debemos tener en cuenta que el sensor en cuestión posee todos sus registros en 16 bits. A continuación se detalla el registro con su configuración adoptada:

#### Registro de Control (Dirección: 0x01)

15	14	13	12	11	10	9	8
OS	R1	R2	FT1	FT2	POL	CMP/INT	SD

Bits 7-0 reservados para futuros usos.

OS: Modo de Disparo único;

0: Operación normal (Modo default);

1: Modo de medición única (Solo válido en modo shutdown).

R1:R0: Resolución de conversión;

00: 9 bits (Resolución de 0,5 [°C]);

01: 10 bits (Resolución de 0,25 [°C]);

10: 11 bits (Resolución de 0,125 [°C]);

11: 12 bits (Resolución de 0,0625 [°C]).

FT1:FT0: Tolerancia de la cola de datos;

00: Alarma ante un desborde;

01: Alarma ante dos desbordes consecutivos;

10: Alarma ante cuatro desbordes consecutivos;

11: Alarma ante seis desbordes consecutivos.

POL: Estado del pin de alarma;

0: Alarma por estado bajo (Pin ALERT);

1: Alarma por estado alto (Pin ALERT).

CMP/INT: Alarma;

0: Modo por comparación;

1: Modo por interrupción.

SD: Modo Apagado;

0: Sensor activado;

1: Sensor desactivado.

### Configuración asignada:

7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1

Mediante la configuración anterior, el sensor se encuentra en el modo de medición única arrojando mediciones con una resolución de 12 bits.

### 2.3.3.3- Descripción del Registro de Datos

El registro de datos del sensor de temperatura se compone como un registro de 16 bits de solo lectura. La parte entera de la medición se encuentra almacenada como un dato en complemento a dos en los bits 15-8, mientras que los bits 7-4 se encargan de almacenar la parte decimal de la medición realizada.

### Registro de Datos (Dirección: 0x00)

Upper Byte								Lower Byte							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Signo	TD	TD	TD	TD	TD	TD	TD	TD	TD	TD	TD	0	0	0	0

### 2.3.3.4- Proceso de Lectura

El proceso de lectura del sensor se efectuó según el diagrama de flujo de la figura N°2.3.

Dicho diagrama de flujo se puede reducir al de la figura N°2.4. Es decir se reduce a la ejecución de las rutinas de Inicialización y Lectura del Sensor AT30TS75.

### 2.3.3.4.1- Rutina de Inicialización del Sensor de Temperatura AT30TS75

```
void init_AT30TS75 (void)
{
    #define SLAVE_I2C_GENERIC_RETRY_MAX 1
    uint8_t timeOut;

    uint8_t configuracion[2] = {
        0x01,
        0b11100001
    };

    void I2C_Initialize(void);

    I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;

    timeOut = 0;
    while(status != I2C_MESSAGE_FAIL)
    {
        I2C_MasterWrite(configuracion,2,SLV_ADD_TEMP,&status);
    }
}
```

```

while(status == I2C_MESSAGE_PENDING);
if (status == I2C_MESSAGE_COMPLETE)
    break;

if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
    break;
else
    timeOut++;
}

```

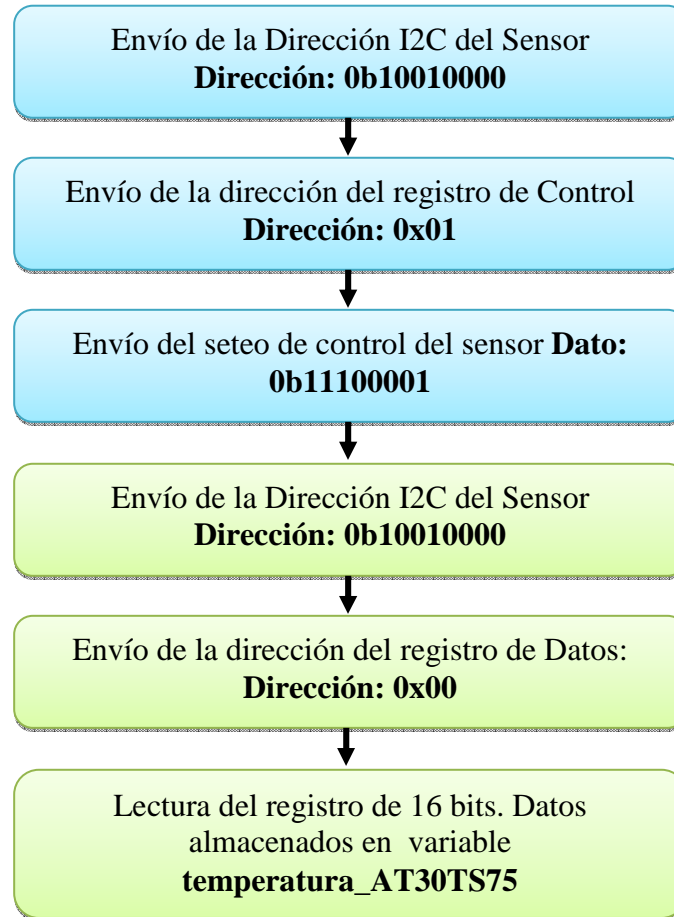


Figura Nº 2.3- Diagrama de Flujo General.

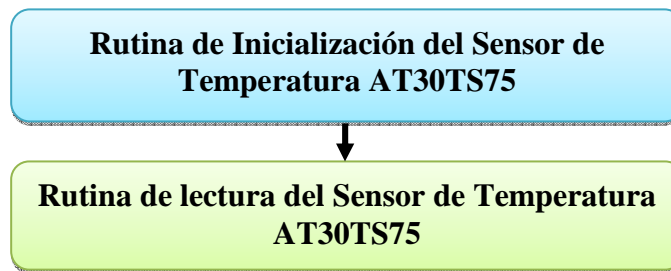
#### 2.3.3.4.2- Rutina de lectura del Sensor de Temperatura AT30TS75

```

void read_AT30TS75(void)
{
    I2C_MESSAGE_STATUS status;
    I2C_MasterWrite(0x00,1,SLV_ADD_TEMP,&status);
    while(status == I2C_MESSAGE_PENDING);
    _delay(1000);
    I2C_MasterRead(temperatura_AT30TS75,4,SLV_ADD_TEMP,&status);
}

```





**Figura Nº 2.4- Diagrama de Flujo Reducido.**

Durante el proceso de lectura se obtuvieron los siguientes valores almacenados en el registro de datos del sensor:

Upper Byte								Lower Byte							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0

Análisis de la parte entera: 0b00011010 = 26 [°C];

Análisis de la parte decimal: 0b0011 = 3 x 0.0625 [°C] = 0,1875 [°C].

De lo anterior resulta una temperatura de 26,1875 [°C]

### 2.3.4- Sensor de Presión Atmosférica MPL3115A2

#### 2.3.4.1- Dirección I2C del Sensor

Con la finalidad de establecer la comunicación entre el Microcontrolador y el Sensor de Presión Atmosférica mediante sus interfaces I2C, dicho sensor posee la siguiente dirección: 0b10010000, 0xC0 (Hexadecimal). Dicha dirección de memoria deberá ser antepuesta a todo mensaje transmitido desde el Microcontrolador hacia el Sensor en cuestión.

#### 2.3.4.2- Descripción de los Registros de Control

Para efectuar la puesta en marcha del Barómetro, inicialmente se realizó la configuración de los registros de control asociados al sensor. A continuación se listan los registros del sensor con su configuración adoptada:

#### PT\_DATA\_CFG (Dirección: 0x13)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	DREM	PDFE	TDEFE
W								

#### Configuración asignada:

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1



	DRDY	FIFO	PW	TW	PTH	TTH	PCHG	TCHG
--	------	------	----	----	-----	-----	------	------

**Configuración asignada:**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

El sensor fue configurado de forma tal de entregar una lectura de Presión y Temperatura cada vez que se lo interrogaba, por lo cual, todas las interrupciones fueron deshabilitadas. Los registros de control son configurados durante la rutina de inicialización y activación del sensor.

**2.3.4.3- Descripción de los Registros de Datos**

La lectura de Presión Barométrica es almacenada en pascales utilizando para tal fin tres registros de 8 bits cada uno, del total de los 24 bits, solo 20 de ellos son utilizados para almacenar el valor de la presión medida. Los primeros 18 bits almacenados en OUT\_P\_MSB, OUT\_P\_CSB y los bits 7-6 del registro OUT\_P\_LSB. Por otro lado, los bits en posición 5 y 4 del registro anterior, representan la parte decimal de la presión medida.

En cuanto a la medición de la temperatura, esta se compone de un dato de 12 bits en complemento a dos almacenados en grados centígrados. Los 8 bits correspondientes al registro OUT\_T\_MSB representan la parte entera de la temperatura, mientras que, los 4 bits almacenados en las posiciones 7-4 del registro OUT\_T\_LSB, representan la parte decimal de la medición realizada. A continuación se detalla el formato de los registros anteriormente mencionados:

**OUT\_P\_MSB (Dirección: 0x01)**

	7	6	5	4	3	2	1	0
R	PDD19	PDD18	PDD17	PDD16	PDD15	PDD14	PDD13	PDD12
W								

**OUT\_P\_CSB (Dirección: 0x02)**

	7	6	5	4	3	2	1	0
R	PDD11	PDD10	PDD9	PDD8	PDD7	PDD6	PDD5	PDD4
W								

**OUT\_P\_LSB (Dirección: 0x03)**

	7	6	5	4	3	2	1	0
R	PDD3	PDD2	PDD1	PDD0	0	0	0	0
W								

**OUT\_T\_MSB (Dirección: 0x04)**

	7	6	5	4	3	2	1	0
R	TDD11	TDD10	TDD9	TDD8	TDD7	TDD6	TDD5	TDD4
W								

**OUT\_T\_LSB (Dirección: 0x05)**

	7	6	5	4	3	2	1	0
R	TDD3	TDD2	TDD1	TDD0	0	0	0	0
W								

La lectura de los registros anteriores se efectúa durante la rutina de lectura del sensor

**2.3.4.4- Proceso de Lectura**

El proceso de lectura del sensor se efectuó según se detalla en el diagrama de flujo de la figura N°2.5.

Los mismos se reducen en la ejecución de las rutinas de inicialización, activación y lectura del sensor de presión atmosférica MLP3115A2, figura N°2.6.

Durante el proceso de lectura se obtuvieron los siguientes valores almacenados en los registros de datos del sensor:

**OUT\_P\_MSB (Dirección: 0x01)**

	7	6	5	4	3	2	1	0
R	0	1	1	0	0	0	1	0

**OUT\_P\_CSB (Dirección: 0x02)**

	7	6	5	4	3	2	1	0
R	0	1	0	0	0	1	1	1

**OUT\_P\_LSB (Dirección: 0x03)**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0

**OUT\_T\_MSB (Dirección: 0x04)**

	7	6	5	4	3	2	1	0
R	0	0	0	1	1	0	0	1

**OUT\_T\_LSB (Dirección: 0x05)**

	7	6	5	4	3	2	1	0
R	1	1	0	0	0	0	0	0

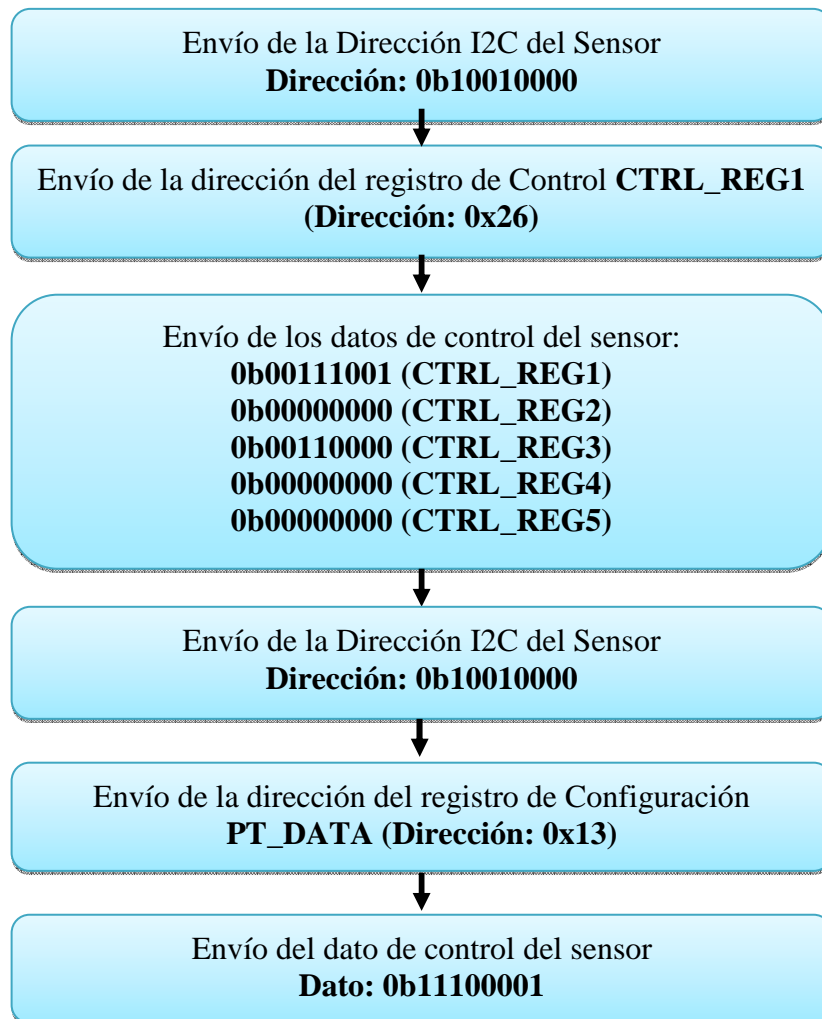
Tomando los 16 bits correspondientes a los registros OUT\_P\_MSB - OUT\_P\_CSB y los bits 7-6 del registro OUT\_P\_LSB, compondremos la parte entera, mientras que, los bits 5-4 del registro anterior, integrarán la parte decimal de la presión medida en pascales.

Parte Entera: 11000100100011100 = 100636 [pa] = 1006,36 [hpa];

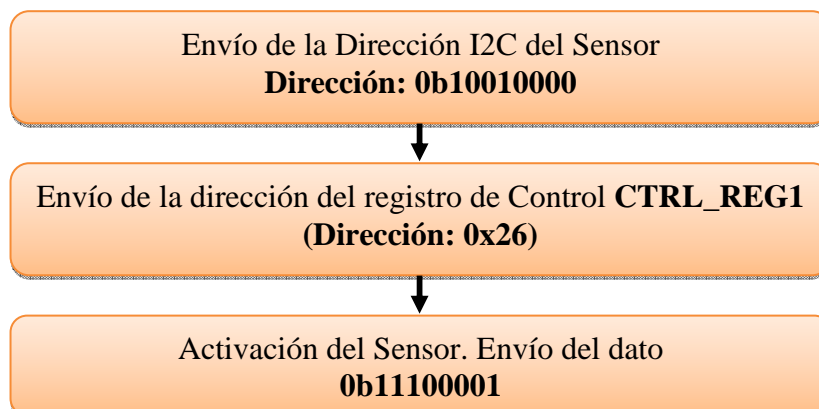
Parte Decimal: 00 = 0,00 [pa];

De lo anterior resulta una presión medida de 1006,36 [hpa]

En cuanto a la temperatura, el resultado de la medición arrojó un valor de 25 °C.



**Figura 2.5 parte 1. Inicialización del Sensor.**



**Figura 2.5 parte 2. Activación del Sensor.**

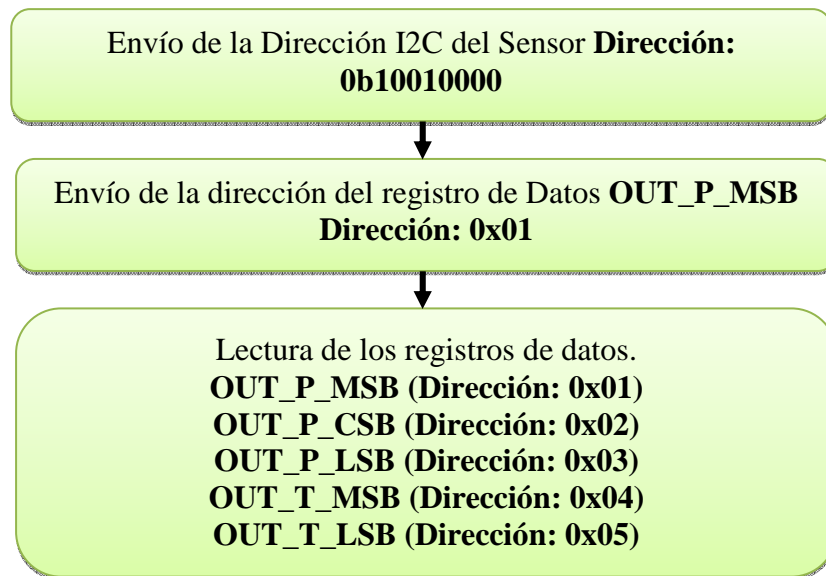


Figura 2.5 parte 2. Lectura del Sensor.

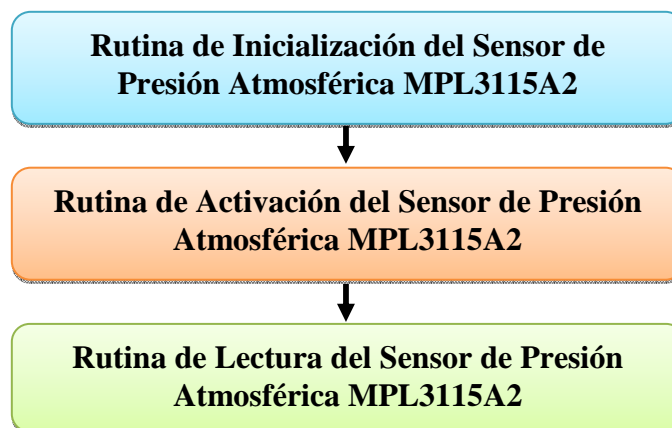


Figura 2.6. Diagrama reducido.

#### 2.3.4.5- Rutina de Inicialización del Sensor de Presión Atmosférica MPL3115A2

```

uint8_t init_MPL3115(void)
{
#define SLAVE_I2C_GENERIC_RETRY_MAX 1
uint8_t timeOut;
MPL3115A2_enabled = 1;

uint8_t registros_control[6] = {0x26,
                                0b00111001, //CTRL_REG1
                                0b00000000, //CTRL_REG2
                                0b00110000, //CTRL_REG3
                                0b00000000, //CTRL_REG4
                                0b00000000 //CTRL_REG5
                                };
uint8_t PT_DATA_config[2] = {0x13,

```

```
0b00000111
};
```

```
void I2C_Initialize(void);
I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;
timeOut = 0;
while(status != I2C_MESSAGE_FAIL)
{
    I2C_MasterWrite(registros_control,6,SLV_ADD_PRES,&status);
    while(status == I2C_MESSAGE_PENDING);
    if (status == I2C_MESSAGE_COMPLETE)
        break;
    if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
        break;
    else
        timeOut++;
}
```

```
I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;
timeOut = 0;
while(status != I2C_MESSAGE_FAIL)
{
    I2C_MasterWrite(PT_DATA_config,2,SLV_ADD_PRES,&status);
    while(status == I2C_MESSAGE_PENDING);
    if (status == I2C_MESSAGE_COMPLETE)
        break;
    if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
        break;
    else
        timeOut++;
}
return MPL3115A2_enabled;
}
```

#### 2.3.4.6- Rutina de Activación del Sensor de Presión Atmosférica MPL3115A2

```
uint8_t active_MPL3115 (void)
{
    #define SLAVE_I2C_GENERIC_RETRY_MAX    1
    uint8_t timeOut;

    MPL3115A2_enabled = 1;

    uint8_t registro_control[2] = {0x26,
                                   0b00111001,
                                   };

    void I2C_Initialize(void);
    I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;
    timeOut = 0;
    while(status != I2C_MESSAGE_FAIL)
    {
        I2C_MasterWrite(registro_control,2,SLV_ADD_PRES,&status);
        while(status == I2C_MESSAGE_PENDING);
        if (status == I2C_MESSAGE_COMPLETE)
```

```

    break;
    if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
        break;
    else
        timeOut++;
}
return MPL3115A2_enabled;
}

```

#### 2.3.4.7- Rutina de Lectura del Sensor de Presión Atmosférica MPL3115A2

```

void read_MPL3115(void)
{
    uint8_t direccion[1];
    direccion[0]=0x01;

    I2C_MESSAGE_STATUS status;
    I2C_MasterWrite(direccion,1,SLV_ADD_PRES,&status);
    while(status == I2C_MESSAGE_PENDING);
    I2C_MasterRead(datos_MPL3115,5,SLV_ADD_PRES,&status);
}

```

#### 2.3.5- Sensor de luz ambiente BH1721FVC

##### 2.3.5.1- Inicialización

Este sensor posee dos rangos de medición, según la cual tendrá determinado tiempo de medición y resolución, llamado H-mode o L-mode.

El H-Resolution mode puede medir un rango reducido de Lux, reduciendo el rango máximo a 0 - 8191Lux, pero su resolución aumenta a 1Lux. No obstante esto el tiempo de medición aumenta a 120ms.

Por otro lado el L-Resolution mode, si bien puede medir en todo el rango de 0 - 65528Lux en tan solo 16ms, la resolución se ve reducida a 8Lux.

Para mayor simplicidad de programación y debido a que los tiempos de lectura en nuestra aplicación no son importantes, hemos seleccionado el modo de Auto-Resolución, en el cual, luego de 16ms de iniciada la aplicación, el propio dispositivo selecciona el modo de resolución. De esta forma, si la lectura inicial es mayor o igual a 4000Lux se coloca en modo L-Resolution y si es menor se coloca en modo H-Resolution.

Para esto el Maestro debe enviar la dirección de escritura del sensor y luego el OPCODE 0b00010000 correspondiente al modo seleccionado.

##### 2.3.5.2- Función de configuración inicial

En el firmware de prueba se le asignó el nombre SLV\_ADD\_LUZ (slave address luz).

```
#define SLV_ADD_LUZ    0x46// Address luxómetro 0100.0110
```

```
void init_light (void); //declaración función de inicialización luxómetro BH1721FVC
```

```
void init_light (void) // definición función de inicialización de luxómetro BH1721FVC
```

```

{
i2c_Open(SLV_ADD_LUZ,I2C_WRITE); // BH1721FVC - Slave Add for Write
i2c_PutByte(0x10);           // auto resolution mode OPCODE

```



```
i2c_Stop ();  
}
```

### 2.3.5.3- Lectura de la medición

El Master debe enviar al Bus la dirección de lectura del sensor y esperar el ACK del sensor. Posteriormente el mismo envía la parte de alta de la medición y el Master responde con un ACK, esperando para leer la parte baja de la medición. Con un NACK y la condición de Stop se indica que la lectura ha finalizado.

### 2.3.5.4- Conversión de la medición a Lux

La medición es registrada en 16bits, por lo que es necesario realizar un cálculo adicional para convertir a la unidad Lux el valor binario obtenido.

Para esto se procede como en el ejemplo siguiente:

Byte alto = "10000011"

Byte bajo = "10010000"

Luego  $(2^{15} + 2^9 + 2^8 + 2^7 + 2^4) / 1.2 = 28067$  [Lux]

Notar además que el sensor tiene la posibilidad de realizar un ajuste en el caso de que la medición sea afectada por una ventana óptica. Para esto se sugiere referirse a la hoja de datos del mismo en el caso de llegar a utilizar este elemento.

### 2.3.5.5- Función de lectura de la medición

```
void read_light (void); //definición de función lectura de luxómetro BH1721FVC
```

```
void read_light (void) //declaración de función lectura de luxómetro BH1721FVC
```

```
{  
i2c_Open(SLV_ADD_LUZ,I2C_READ); // BH1721FVC - Slave Add for Read  
LiH= i2c_GetByte(I2C_MORE); // read LiH  
LiL= i2c_GetByte(I2C_LAST); // read LiL  
i2c_Stop();  
}
```

## 2.3.6- Sensor de Humedad Relativa y Temperatura Si7005

### 2.3.6.1- Dirección I2C del Sensor

Con la finalidad de establecer la comunicación entre el Microcontrolador y el Sensor de Humedad Relativa y Temperatura mediante sus interfaces I2C, dicho sensor posee la siguiente dirección: **0b01000000, 0x40 (Hexadecimal)**. Dicha dirección de memoria deberá ser antepuesta a todo mensaje transmitido desde el Microcontrolador hacia el Sensor en cuestión.

### 2.3.6.2- Descripción del Registro de Control

Para efectuar la puesta en marcha Sensor de Humedad Relativa y Temperatura, inicialmente se realizó la configuración del registro **CONFIG**. Dicha configuración se encuentra diferenciada entre lectura de la Humedad Relativa o Temperatura. A continuación se detalla el registro y la configuración asociada a la lectura de cada variable:



**DATAI (Dirección: 0x02)**

	7	6	5	4	3	2	1	0
R	HUM3	HUM2	HUM1	HUM0	0	0	0	0
W								

El Valor Leído luego de una lectura debe ser convertido a Porcentaje de Humedad Relativa [% HR] mediante el uso de la siguiente fórmula:

$$\% HR = \frac{\text{Valor Leído}}{16} - 24$$

**2.3.6.3.2- Lectura de Temperatura**

**DATAh (Dirección: 0x01)**

	7	6	5	4	3	2	1	0
R	TEMP1 3	TEMP1 2	TEMP1 1	TEMP1 0	TEMP 9	TEMP 8	TEMP 7	TEMP 6
W								

**DATAI (Dirección: 0x02)**

	7	6	5	4	3	2	1	0
R	TEMP5	TEMP4	TEMP3	TEMP2	TEMP1	TEMP0	0	0
W								

El Valor Leído luego de una lectura debe ser convertido a Grados Centígrados [° C] mediante el uso de la siguiente fórmula:

$$^{\circ} C = \frac{\text{Valor Leído}}{32} - 50$$

### 2.3.6.4- Proceso de Lectura de la Humedad Relativa

El proceso de lectura de la Humedad Relativa se efectuó según se detalla en el diagrama de flujo de la figura N°2.7:

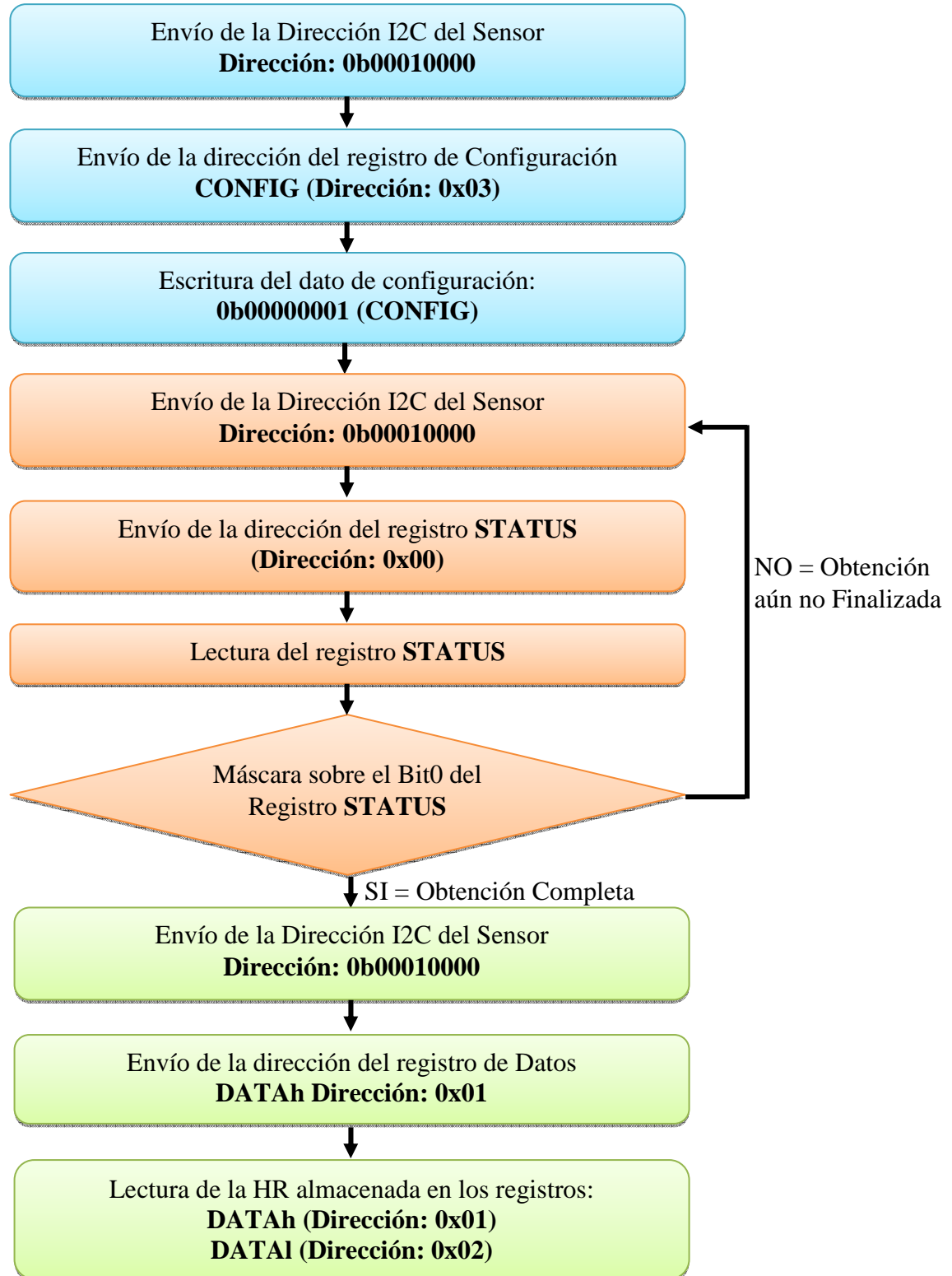


Figura N°2.7. Proceso de Lectura de la Humedad Relativa

### 2.3.6.5- Proceso de Lectura de la Temperatura

El proceso de lectura de la Temperatura se efectuó según se detalla en el diagrama de flujo de la figura N°2.8.

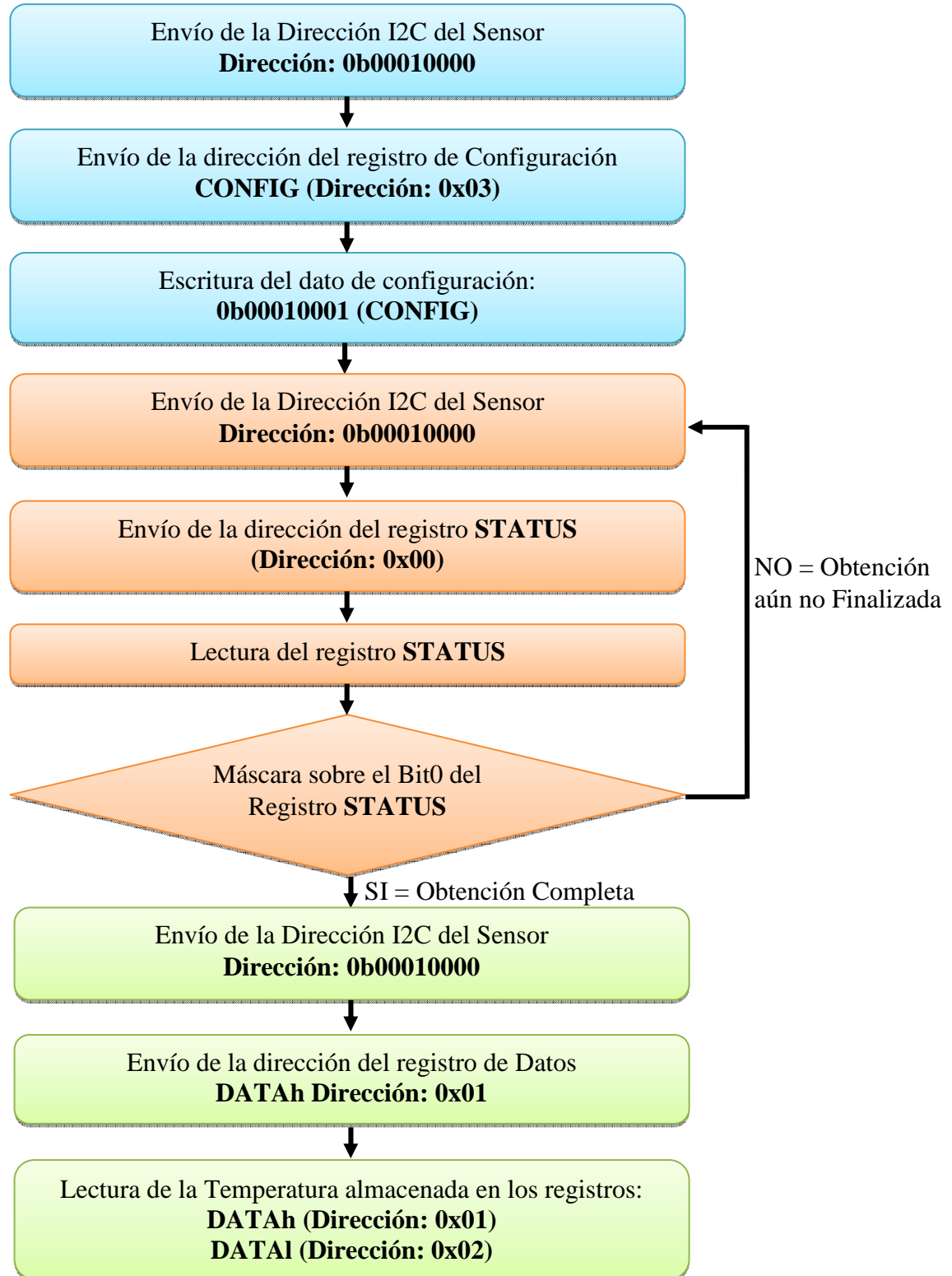


Figura N°2.8. Proceso de Lectura de la Temperatura

El diagrama de flujo correspondiente a la lectura de la Humedad Relativa se reduce en la ejecución de las rutinas de Inicialización, y Lectura de la Humedad Relativa, figura N°2.9.

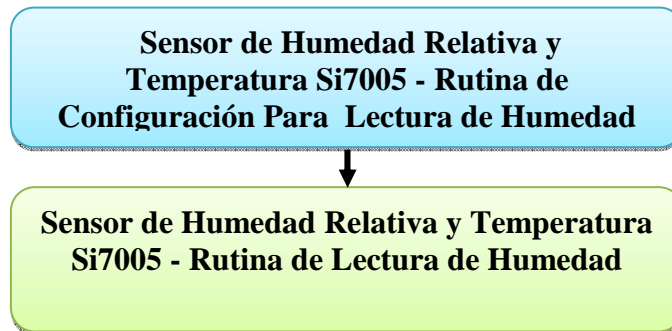


Figura N°2.9. Proceso reducido de Lectura de la Humedad

Por otro lado, el diagrama de flujo correspondiente a la lectura de la Temperatura se reduce en la ejecución de las rutinas de Inicialización, y Lectura de la Temperatura, figura N°2.10.

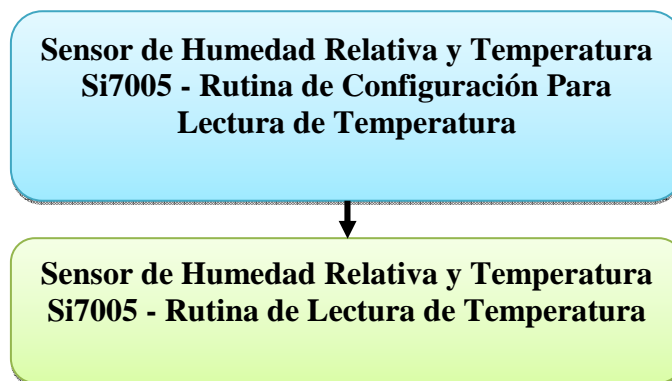


Figura N°2.10. Proceso reducido de Lectura de la Temperatura

#### 2.3.6.6- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Configuración Para Lectura de Humedad

```

uint8_t init_Hum_Si7005(void)
{
#define SLAVE_I2C_GENERIC_RETRY_MAX 1
uint8_t timeOut;
Si7005_enabled = 1;

uint8_t Set_Si7005[2] = {0x03,
                        0b00000001,
                        };

void I2C_Initialize(void);
I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;
timeOut = 0;

while(status != I2C_MESSAGE_FAIL)
  
```

```

{
  I2C_MasterWrite(Set_Si7005,2,SLV_ADD_HUM,&status);
  while(status == I2C_MESSAGE_PENDING);
  if (status == I2C_MESSAGE_COMPLETE)
    break;
  if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
    break;
  else
    timeOut++;
}

return Si7005_enabled;
}

```

### 2.3.6.7- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Configuración Para Lectura de Temperatura

```

uint8_t init_Temp_Si7005(void)
{
#define SLAVE_I2C_GENERIC_RETRY_MAX 1
  uint8_t timeOut;
  Si7005_enabled = 1;

  uint8_t Set_Si7005[2] = {0x03,
                          0b00010001,
                          };

  void I2C_Initialize(void);
  I2C_MESSAGE_STATUS status = I2C_MESSAGE_PENDING;
  timeOut = 0;
  while(status != I2C_MESSAGE_FAIL)
  {
    I2C_MasterWrite(Set_Si7005,2,SLV_ADD_HUM,&status);
    while(status == I2C_MESSAGE_PENDING);
    if (status == I2C_MESSAGE_COMPLETE)
      break;
    if (timeOut == SLAVE_I2C_GENERIC_RETRY_MAX)
      break;
    else
      timeOut++;
  }
  return Si7005_enabled;
}

```

### 2.3.6.8- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Lectura de Humedad

```

void read_hum_Si7005 (void)
{
  uint8_t direccion [1];
  direccion [0]=0x00;

  while (status_Si7005 [0] & 0b00000001)

```

```

{
  I2C_MESSAGE_STATUS status;
  I2C_MasterWrite(direccion, 1, SLV_ADD_HUM, &status);
  while(status == I2C_MESSAGE_PENDING);
  I2C_MasterRead(status_Si7005, 1, SLV_ADD_HUM, &status);
}

direccion [0]=0x01;
I2C_MESSAGE_STATUS status;
I2C_MasterWrite(direccion, 1, SLV_ADD_HUM, &status);
while(status == I2C_MESSAGE_PENDING);
I2C_MasterRead(humedad_Si7005, 2, SLV_ADD_HUM, &status);
}

```

### 2.3.6.9- Sensor de Humedad Relativa y Temperatura Si7005 - Rutina de Lectura de Temperatura

```

void read_temp_Si7005 (void)
{
  uint8_t direccion[1];
  direccion[0]=0x00; // Dirección del Registro STATUS

  while (status_Si7005 [0] & 0b00000001)
  {
    I2C_MESSAGE_STATUS status;
    I2C_MasterWrite(direccion, 1, SLV_ADD_HUM, &status);
    while(status == I2C_MESSAGE_PENDING);
    I2C_MasterRead(status_Si7005, 1, SLV_ADD_HUM, &status);
  }

  direccion[0]=0x01;
  I2C_MESSAGE_STATUS status;
  I2C_MasterWrite(direccion, 1, SLV_ADD_HUM, &status);
  while(status == I2C_MESSAGE_PENDING);
  I2C_MasterRead(temperatura_Si7005, 2, SLV_ADD_HUM, &status);
}

```

## 2.4- Conclusiones

Se cumplió parcialmente el objetivo de puesta en marcha de la placa. La fuente de alimentación, el sensor de temperatura y de presión se fueron puestos en marcha y se verifico su despeño, si bien el transreceptor también se tratara en más detalle la siguiente sección. El sensor de humedad relativa y temperatura (Si7005) no llego a ponerse en marcha por cuestiones de tiempo, mientras que el sensor de intensidad luminosa presento severos problemas en el momento del montaje.

Es importante destacar la experiencia adquirida por la persona responsable de la puesta en marcha de dicho hardware.



### 3- Transceptor de la microestación

Si bien el transceptor forma parte de la microestación, se presenta su puesta en marcha en una sección independiente por la trascendencia que este implica.

Tal como se

#### 3.1- Tipos de protocolos de comunicación

Se analizaron los distintos stacks (pilas) de firmware disponibles soportados por la plataforma desarrollada, con el fin de abordar una solución de una red RF autoconfigurable de bajo nivel. Entre ellos podemos nombrar:

- Lightweigh Mesh.
- BitCloud.
- Contiki

A continuación se describe brevemente las funcionalidades de cada uno de ellos

##### 3.1.1-Lightweight Mesh

Este stack permite implementar un protocolo de comunicaciones de baja potencia para redes mesh, el mismo es propietario de ATMEL. Esta diseñado para trabajar con todos los transceptores y SoCs de ATMEL que trabajen con el estandar IEEE 802.15.4.

Sus características son:

- Simplicity of configuration and use
- Up to 65535 nodes in one network (theoretical limit)
- Up to 65535 separate PANs on one channel
- 15 independent application endpoints
- No dedicated node is required to start a network
- No periodic service traffic occupying bandwidth
- Two distinct types of nodes:
- Routing (network address < 0x8000)
- Non-routing (network address ≥ 0x8000)
- Once powered on node is ready to send and receive data; no special joining procedure is required
- No child-parent relationship between the nodes
- Non-routing nodes can send and receive data to/from any other node (including non-routing nodes), but they
- will never be used for routing purposes
- Route discovery happens automatically if route to the destination is not known
- Routing table is updated automatically based on the data from the received and transmitted frames
- Optional support for AODV routing
- Optional support for multicast communication
- Duplicate frames (broadcast or multipath unicast) are rejected
- Small footprint (8KB of Flash and 4KB of RAM for a typical application)

Este stack/protocolo permite implementar redes del tipo mesh como la que se ve en la figura 3.1. En ella se destacan dos tipos de dispositivos, routers y no routers.

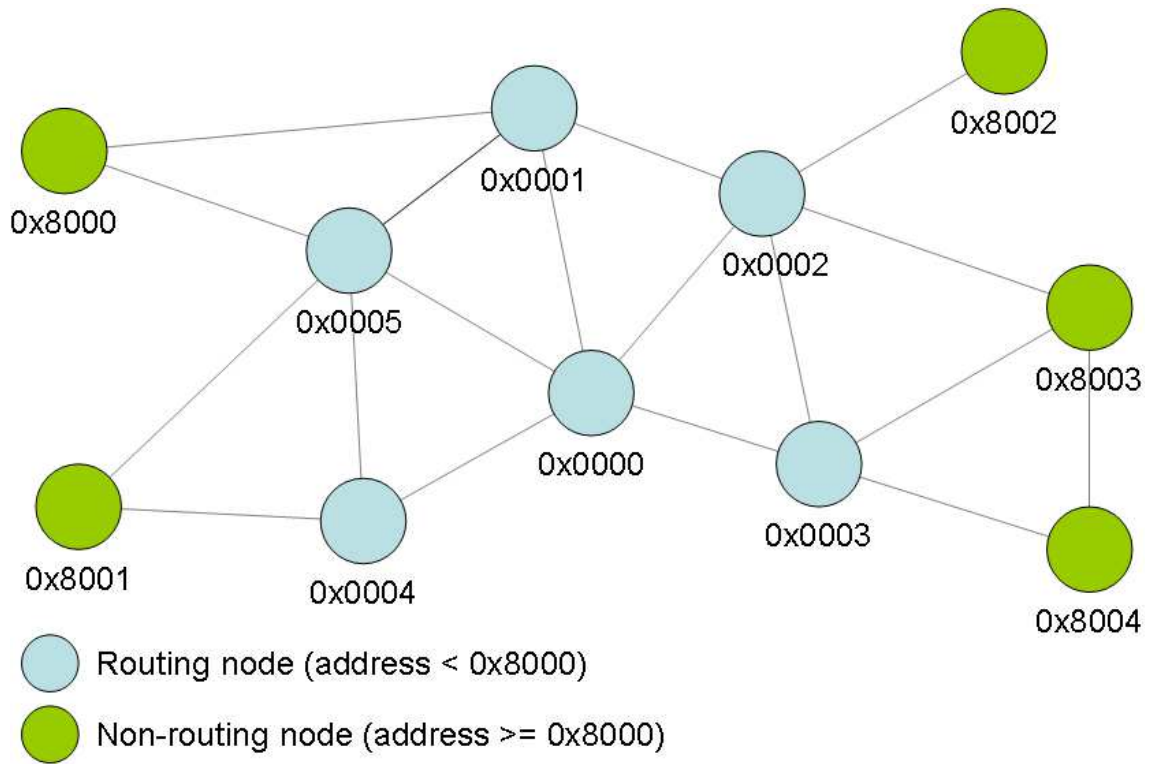


Figura 3.1. Topología de Red.

La arquitectura del stack se muestra en la figura 3.2, en esta se puede apreciar distintos niveles lógicos, para los cuales es provisto un conjunto de APIs.

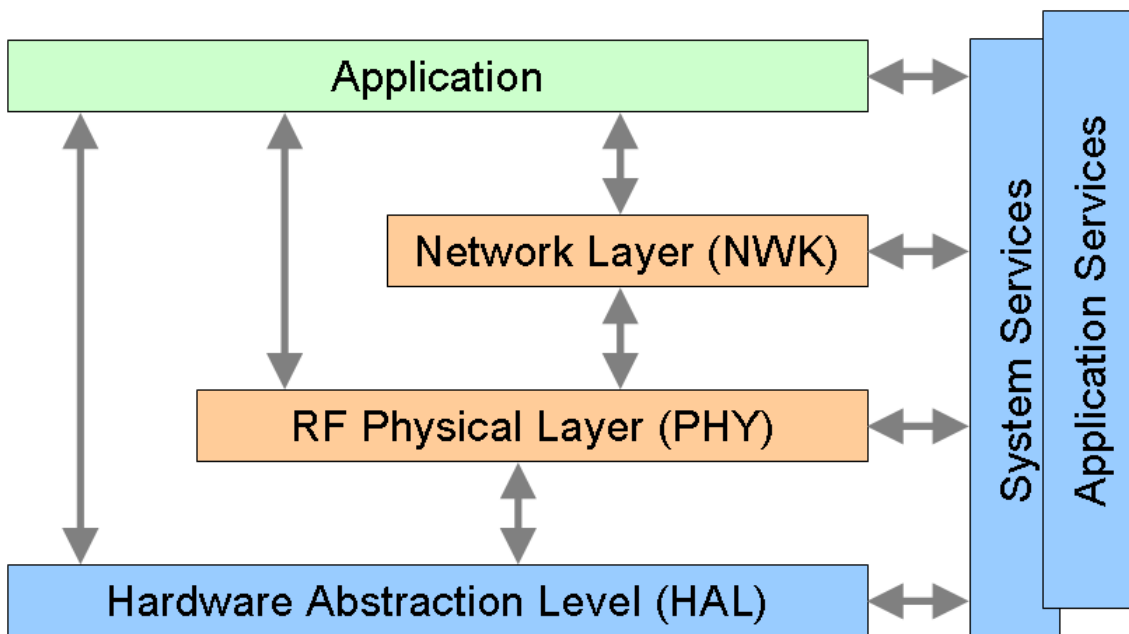


Figura 3.2. Arquitectura del stack.

Las ventajas que posee el uso de este stack/protocolo es su simpleza y su reducido tamaño de código. Como desventaja podemos nombrar que el mismo es propietario, lo

que no va permitir interoperabilidad con otros productos del mercado, así como también un menor número de funcionalidades. Más información sobre este stack/protocolo se puede encontrar en [34].

### 3.1.2- BitCloud

Este stack permite implementar el protocolo Zigbee PRO de comunicaciones completo. Pudiendo implementar aplicaciones con los perfiles ZigBee Home Automation (ZHA), ZigBee Light Link (ZLL), y dispositivos propietarios ZigBee. Esta diseñado para trabajar con todos los transeptores y SoCs de ATMEL que trabajen con el estandar IEEE 802.15.4.

Las características del stack/protocolo Zigbee así como los las topologías de redes como las funcionalidades de los distintos dispositivos, se detalla claramente en [35]. La figura 3.3 se puede apreciar arquitectura del stack. Como se puede ver aumenta el nivel de complejidad, así como la funcionalidad del mismo. Esto tiene como consecuencia directa el aumento del tamaño del código necesario para implementarlo en nuestro sistema embebido [36].

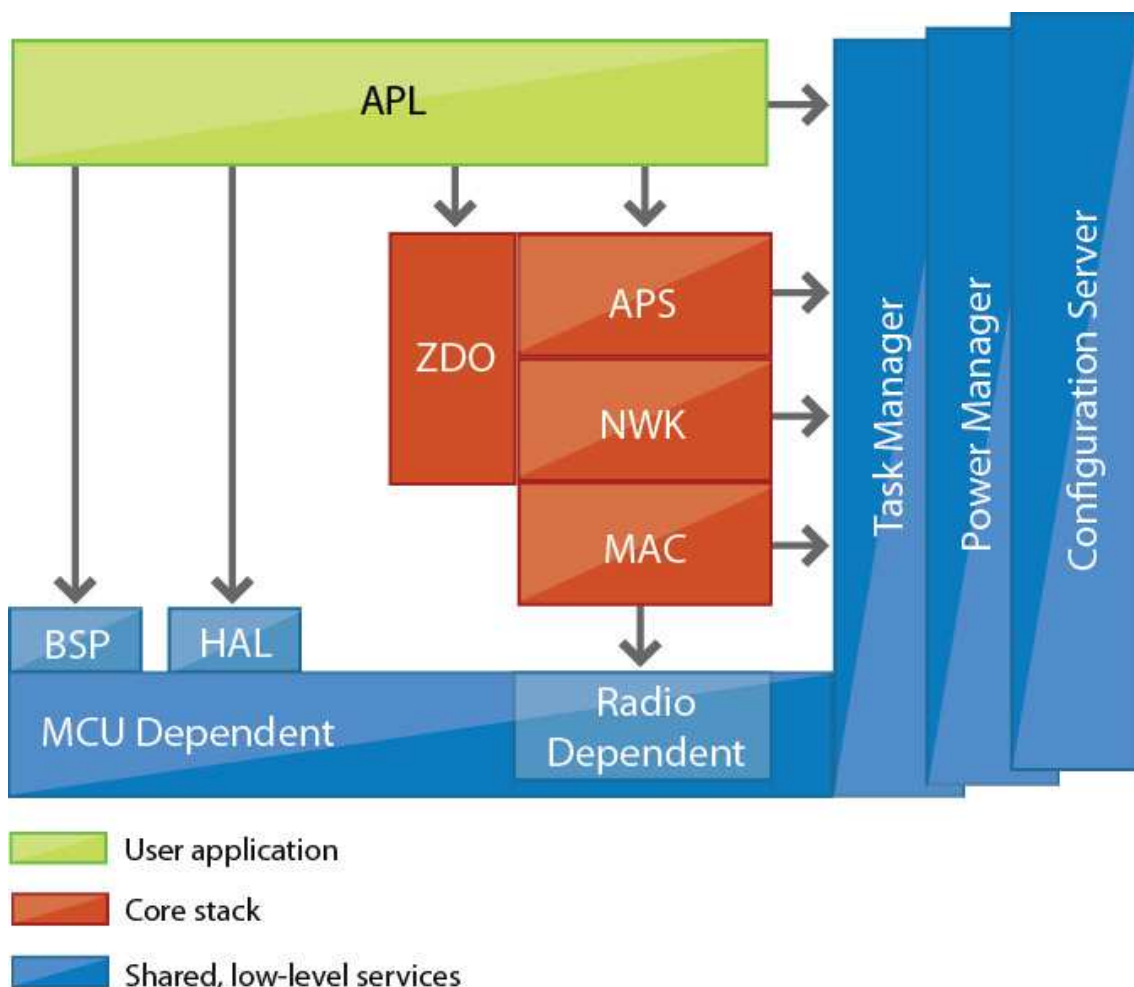


Figura 3.3. Arquitectura del stack Bitcloud

### 3.1.3- Contiki

En dispositivos de cierta entidad y alimentados mediante red eléctrica es posible utilizar TCP/IP sobre redes WiFi o Celulares (GSM/GPRS, 3G, etc), y que el dispositivo implemente los conocidos servicios WEB de tipo REST para su manejo, es decir obtener datos o indicar la ejecución de operaciones sobre los datos a través de la implementación de una interfaz entre sistemas que utilice directamente HTTP.

Sin embargo, uno de los retos a los que nos enfrentamos en la comunicación de los dispositivos es que, en muchas ocasiones, se trata de pequeños sensores alimentados por baterías y enlazados con pasarelas (gateways) más potentes a través de redes radio de baja potencia (redes WSN o LowPAN. Estas redes de radio tienen mayor probabilidad de pérdida de mensajes y para su funcionamiento óptimo precisan que los mensajes sean lo más cortos posibles. Como consecuencia la implementación de servicios WEB de tipo REST resulta ineficiente y desaconsejable en estos entornos.

Surge entonces, y en vista de cubrir esta aplicación, 6LOWPAN (IPv6 over Low power Wireless Personal Area Networks), definido por la organización de estándares de protocolos de Internet IETF (Internet Engineering Task Force). Este es un estándar que posibilita el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4 [77]. Hace posible que dispositivos como los nodos de una red inalámbrica puedan comunicarse directamente con otros dispositivos IP. La figura 3.4 nos muestra una red típica.

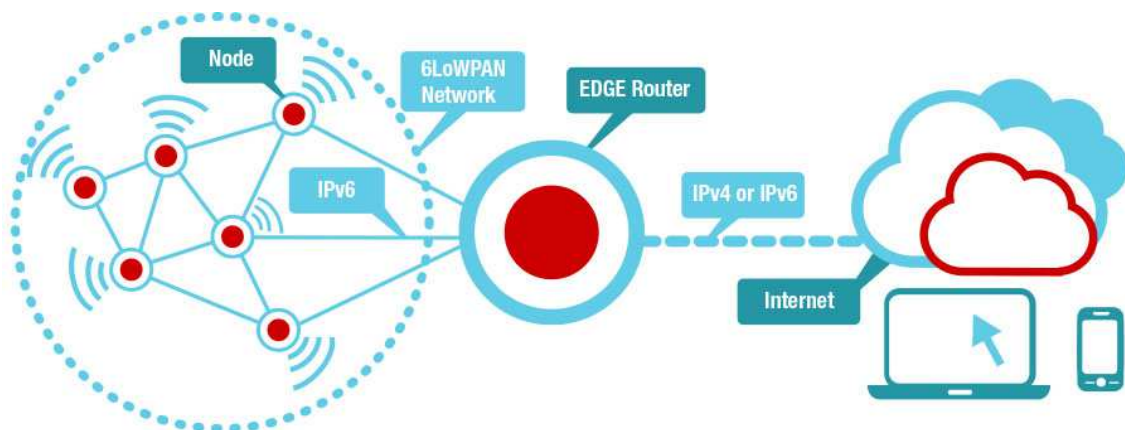


Figura N°3.4. Red 6LowPAN.

Este protocolo se puede encontrar implementado en un sistema operativo de tiempo denominado "Contiki" [37].

Contiki fue creado por Adam Dunkels en 2002 y ha sido desarrollado por un equipo mundial de los desarrolladores de Texas Instruments , Atmel , Cisco, ENEA , ETH Zurich , Redwire , RWTH Aachen University , la Universidad de Oxford , SAP, Sensinode , Instituto Sueco de Ciencias de la Computación , ST Microelectronics , Zolertia , y muchos otros.

Contiki es un sistema operativo de código abierto para sistemas en red, con limitaciones de memoria y con un enfoque particular en baja potencia e Internet inalámbrico de dispositivos.

A pesar de proporcionar multitarea y un built-in pila TCP / IP, Contiki sólo necesita unos 10 kb de memoria RAM y 30 kb de ROM. Un sistema típico Contiki tiene memoria en el orden de kb , con un presupuesto de energía del orden de [mW] , velocidad de procesamiento [MHz] , y ancho de banda de comunicación del orden de cientos de [kBaud].

### **3.1.4- Selección del stack**

No se ha profundizado en Contiki, ya que implica el estudio de programación de sistemas embebidos utilizando un sistema operativo de tiempo real (RTOS). Lo cual no es el objetivo en esta etapa del proyecto.

Por lo tanto para la puesta en marcha de las microestaciones se decidió usar el stack LightWeight Mesh, debido a su discriminación de los elementos básicos necesarios de configuración que intervienen para realizar/modificar la comunicación RF.

Este último stack, está compuesto por varias aplicaciones, de las cuales se seleccionó para la puesta en marcha del transceptor la aplicación cuyo nombre es "peer to peer"(p2p).

### **3.2- Pruebas de la etapa de RF**

Si bien al momento de tener la plataforma lista para realizar los ensayos nos enteramos que el producto utilizado se ha discontinuado, siendo reemplazado por una segunda generación no 100% compatible, se decidió continuar las pruebas para validar que los diseños realizados. Para esto se efectuaron las siguientes pruebas:

- Eficiencia del diseño.
- Alcance.

Hemos llamado eficiencia de diseño al análisis entre la potencia máxima que puede transmitir, teóricamente, el dispositivo y lo potencia que se logra transmitir con nuestro diseño.

Para realizar la prueba de eficiencia de diseño, sobre el modulo ATZB-900-B0 que incorpora el transceptor AT86RF212, se empleo el modo de testeo de transmisión continua que incluye, el cual se encuentra detallado en el apéndice A de la hoja de datos [38]. Configurado el canal y la potencia, en nuestro caso 10 dBm, a la que va transmitir se activa el modo. Esto nos permitió realizar las mediciones de potencia que se pueden ver en las figuras N°2.11 y N°2.12.

Según las mediciones realizadas se obtuvo una potencia de 7,3 dBm, lo cual representa un 73% de la potencia máxima del transmisor [38]. Si bien dicho valor no es el máximo, se considera adecuado en función de las limitaciones impuestas por el diseño, como ser: soporte de múltiples frecuencias (900MHz, 2,4GHz), diseño desarrollado en un circuito impreso de solo dos capas y empleo de FR4, de espesor estándar de 1,6mm, como material base de los circuitos impresos.

Si bien no se desarrollo la misma prueba sobre el modulo ATZB-24-B0, ya que el mismo no cuenta con el modo de testeo de transmisión continua, se estima un desempeño similar.

Para las pruebas de alcance se utilizo un firmware de ejemplo entregado por el fabricante de los módulos, el cual permite establecer una conexión punto a punto. El

ejemplo, Peer2peer, se encuentra disponible el paquete BitCloud\_ZIGBIT\_1\_14\_0. Si bien dicho ejemplo fue desarrollado con el entorno ATMEL STUDIO 5, el mismo de migro al ATMEL STUDIO 6.1.2665 Service Pack 1.

Las pruebas fueron realizadas en las instalaciones deportivas de la Universidad Nacional de la Matanza, quien cuenta con pista de atletismo y cancha de futbol una a continuación de la otra, figura N°2.13, permitiendo distancias mayores a los 100m de línea de vista sin obstáculos.

En estas condiciones se logro un alcance del orden de los 70m con el transceptor de 2.4GHz y se cubrió la distancia del estadio de atletismo y la cancha de futbol, aproximadamente 300m, con el transceptor de 900 Mhz.

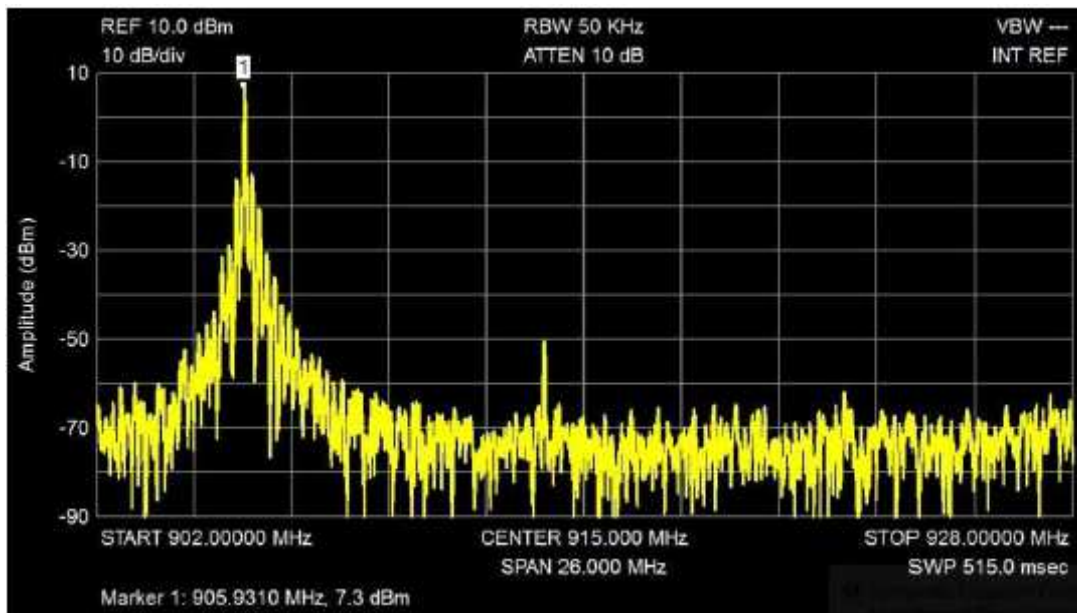


Figura N°2.11.

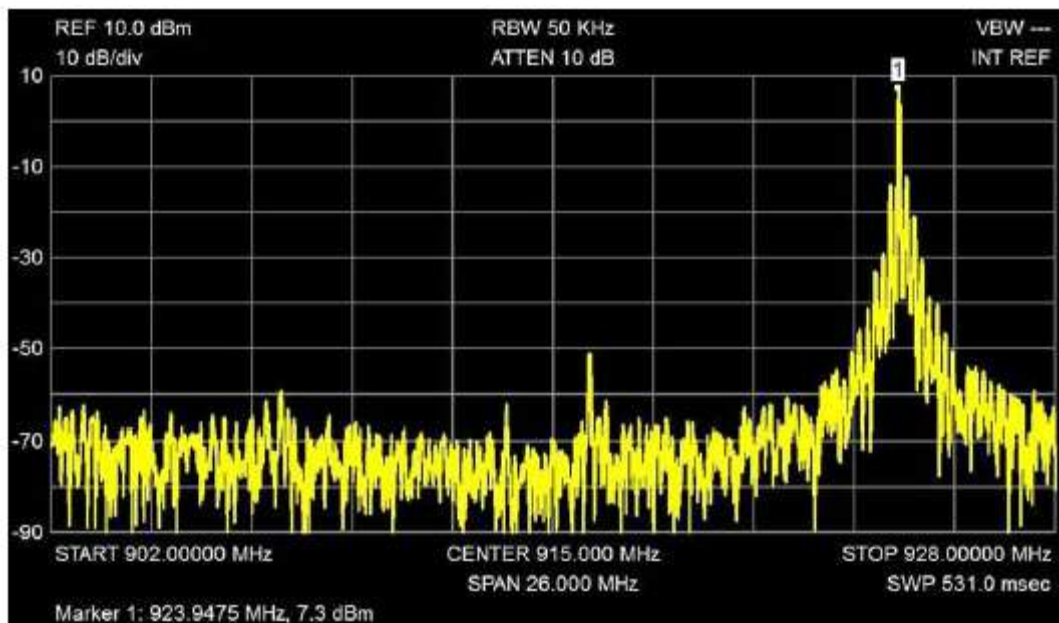


Figura N°2.12.



Figura N°2.13.

### 3.3- Presentación general de la aplicación p2p

A continuación se presenta el código de la aplicación, el cual ha sido documentado en castellano para su mejor interpretación. El mismo pertenece al stack Lightweight Mesh, Release Version: 1.1.1 de fecha Agosto 15, 2013, el ejemplo Peer2Peer.

```

/*
    BREVE DESCRIPCIÓN DE LA APLICACIÓN:

ESTE FIRMWARE, CONSTA DE UN ENLACE PUNTO A PUNTO BIDIRECCIONAL, EN EL CUAL
RECIBE DESDE UNO HASTA UNA TRAMA DE DATOS POR LA UNIDAD UART(CONFIGURADA CON UN
BAUDRATE DE 9600), Y LUEGO ESTE, LO ENVÍA POR RF A OTRA UNIDAD CON LOS MISMOS
DATOS DE CONFIGURACIÓN DE CANAL, PANID, SECURITY KEY, Y DIFERENTE "END POINT".
*/

/*- Includes -----*/
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "config.h"
#include "hal.h"
#include "phy.h"
#include "sys.h"
#include "nwk.h"
#include "sysTimer.h"
#include "halUart.h"

/*- Definitions -----*/
#ifdef NWK_ENABLE_SECURITY
    #define APP_BUFFER_SIZE    (NWK_MAX_PAYLOAD_SIZE - NWK_SECURITY_MIC_SIZE)
#else
    #define APP_BUFFER_SIZE    NWK_MAX_PAYLOAD_SIZE
#endif

/*- Types -----*/

```

```

typedef enum AppState_t
{
    APP_STATE_INITIAL,
    APP_STATE_IDLE,
} AppState_t;

/*- Prototypes -----*/
static void appSendData(void);

/*- Variables -----*/
static AppState_t appState = APP_STATE_INITIAL;
static SYS_Timer_t appTimer;
static NWK_DataReq_t appDataReq;
static bool appDataReqBusy = false;
static uint8_t appDataReqBuffer[APP_BUFFER_SIZE];
static uint8_t appUartBuffer[APP_BUFFER_SIZE];
static uint8_t appUartBufferPtr = 0;

/*- Implementations -----*/

/*****
*****
static void appDataConf(NWK_DataReq_t *req)
{
    appDataReqBusy = false;
    (void)req;
}

/*****
*****
static void appSendData(void)
{
    if (appDataReqBusy || 0 == appUartBufferPtr)
        return;

    memcpy(appDataReqBuffer, appUartBuffer, appUartBufferPtr);

    /* esta función copia lo que hay en el buffer de la UART
    y lo copia al buffer de transmisión de datos rf, para luego ser enviado
    appDataReqBuffer: bus de transmisión de datos rf.
    appUartBuffer: bus de datos de la unidad UART del ATMEGA1281.
    appUartBufferPtr: longitud del dato del bus de la unidad UART del ATMEGA1281.
    */

    appDataReq.dstAddr = 1-APP_ADDR;
    appDataReq.dstEndpoint = APP_ENDPOINT;
    appDataReq.srcEndpoint = APP_ENDPOINT;
    appDataReq.options = NWK_OPT_ENABLE_SECURITY;
    appDataReq.data = appDataReqBuffer;// dato de la uart copiado por la función
memcpy("memory copy")
    appDataReq.size = appUartBufferPtr;// longitud del dato a copiar de la unidad
UART del ATMEGA1281
    appDataReq.confirm = appDataConf;
    NWK_DataReq(&appDataReq);

    appUartBufferPtr = 0;
    appDataReqBusy = true;
}

/*****
*****
void HAL_UartBytesReceived(uint16_t bytes)

```



```

{
    for (uint16_t i = 0; i < bytes; i++)
    {
        uint8_t byte = HAL_UartReadByte();// esta función

        if (appUartBufferPtr == sizeof(appUartBuffer))
            appSendData();

        if (appUartBufferPtr < sizeof(appUartBuffer))
            appUartBuffer[appUartBufferPtr++] = byte;
    }

    /*
    Esta función recibe la información proveniente por la USART, y completa
    las estructuras "appUartBufferPtr" y "appUartBuffer", para luego ser utilizadas
    por la función "appSendData()"
    */
    SYS_TimerStop(&appTimer);
    SYS_TimerStart(&appTimer);
}

/*****
*****
static void appTimerHandler(SYS_Timer_t *timer)
{
    appSendData();
    (void)timer;
}

/*****
*****
/*
Esta función envía por USART los datos recibidos por RF
*/
static bool appDataInd(NWK_DataInd_t *ind)
{
    for (uint8_t i = 0; i < ind->size; i++)
        HAL_UartWriteByte(ind->data[i]);
    return true;
}

/*****
*****
static void appInit(void)
{
    NWK_SetAddr(APP_ADDR);
    NWK_SetPanId(APP_PANID);
    PHY_SetChannel(APP_CHANNEL);
#ifdef PHY_AT86RF212
    PHY_SetBand(APP_BAND);
    PHY_SetModulation(APP_MODULATION);
#endif
    PHY_SetRxState(true);

    NWK_OpenEndpoint(APP_ENDPOINT, appDataInd);

    // Enable RCB_BB RS232 level converter
#ifdef PLATFORM_RCB128RFA1 || defined(PLATFORM_RCB256RFR2)
    DDRD = (1 << 4) | (1 << 6) | (1 << 7);
    PORTD = (0 << 4) | (1 << 6) | (1 << 7);
#endif
}

```

```

#if defined(PLATFORM_RCB231)
    DDRC = (1 << 4) | (1 << 6) | (1 << 7);
    PORTC = (0 << 4) | (1 << 6) | (1 << 7);
#endif

#if defined(PLATFORM_XPLAINED_PRO) && defined(HAL_ATMEGA256RFR2)
    // Enable chip antenna
    DDRG = (1 << 1);
    PORTG = (0 << 1);

    DDRF = (1 << 2);
    PORTF = (1 << 2);
#endif

    appTimer.interval = APP_FLUSH_TIMER_INTERVAL;
    appTimer.mode = SYS_TIMER_INTERVAL_MODE;
    appTimer.handler = appTimerHandler;
}

/*****
*****/
static void APP_TaskHandler(void)
{
    switch (appState)
    {
        case APP_STATE_INITIAL:
        {
            appInit();
            appState = APP_STATE_IDLE;
        } break;

        case APP_STATE_IDLE:
            //leer sensores por I2C(unidad inicializada anteriormente), para luego
            enviar //dichos datos por la rf para recibir por unidad UART(pasar luego por
            driver //CMOS/RS-232)

            break;

        default:
            break;
    }
}

/*****
*****/
int main(void)
{
    SYS_Init();
    HAL_UartInit(9600);

    while (1)
    {
        SYS_TaskHandler();
        HAL_UartTaskHandler();
        APP_TaskHandler();
    }
}

////////////////////////////////////FIN////////////////////////////////////
///

```

Para que ésta aplicación funcione correctamente, es necesario realizar una serie de modificaciones sobre el ejemplo base. Éstas están relacionadas con el punto de envío de trama (denominado "APP\_ADDR"), y con el permiso de envío de ACK, para la detección o avistamiento entre ambos dispositivos.

Los cambios necesarios son:

En el archivo "config.h"

Para el primer dispositivo

```
#define APP_ADDR 0
```

Para el segundo dispositivo

```
#define APP_ADDR 1
```

Lo cual define la dirección de cada uno de los dispositivos. Respecto al permiso de envío de ACK, el cambio se debe realizar en el archivo "Peer2Peer.c". Cambiar:

```
appDataReq.options = NWK_OPT_ENABLE_SECURITY;
```

Por:

```
appDataReq.options = NWK_OPT_ENABLE_SECURITY |  
NWK_IND_OPT_ACK_REQUESTED;
```

### 3.4- Enlace punto a punto Bit Cloud(P2P-BC)

A continuación se presenta el código de la aplicación, el cual ha sido documentado en castellano para su mejor interpretación. El mismo pertenece al stack BitCloud, Release Version: 1.41.0 de fecha Mayo 16, 2012, el ejemplo Peer2Peer.

```
/* BREVE DESCRIPCIÓN DE LA APLICACIÓN:  
  
ESTE FIRMWARE, CONSTA DE UN ENLACE PUNTO A PUNTO BIDIRECCIONAL, EN EL CUAL  
RECIBE DESDE UNO HASTA UNA TRAMA DE DATOS POR LA UNIDAD UART(CONFIGURADA CON UN  
BAUDRATE DE 9600), Y LUEGO ESTE, LO ENVÍA POR RF A OTRA UNIDAD CON LOS MISMOS  
DATOS DE CONFIGURACIÓN DE CANAL, PANID, SECURITY KEY, Y DIFERENTE "END POINT".  
  
ES RELEVANTE DESTACAR QUE LA DIFERENCIA CON EL FIRMWARE CORRESPONDIENTE A ENLACE  
PUNTO A PUNTO DE "LIGHT MESH" SOLO SOPORTA DOS PLATAFORMAS O NIVELES DE  
DIRECCIONES; EN ESTE CASO, ESTE FIRMWARE SOPORTA 8 NIVELES DE DIRECCIONES,  
PERMITIENDO LOS ENLACES PUNTO A PUNTO ENTRE LAS PLATAFORMAS QUE POSEAN LAS  
DIRECCIONES 0-1, 2-3, 4-5, 6-7. ESTO PODRÌA PERMITIR EN PRIMERA INSTANCIA, LA  
COMUNICACIÓN DE TODAS LAS PLATAFORMAS ENTRE SI REALIZANDO CONFIGURACIONES DE LAS  
DIRECCIONES DE CADA UNO DE LOS DISPOSITIVOS.  
*/  
  
/*****  
 \file Peer2Peer.c  
  
*****/  
/*****  
/*****  
 Secciones Incluidas  
*****/  
#include <types.h>
```

```

#include <taskManager.h>
#include <configServer.h>
#include <appTimer.h>
#include <zdo.h>
#include <peer2peer.h>
#include <serialInterface.h>

#if BSP_ENABLE_RS232_CONTROL == 1
#include "rs232Controller.h"
#endif // habilitación rs232

/*****
Sección de definiciones
*****/

/*****
Sección de declaración de variables
*****/
// variables relacionadas con la red
static uint16_t nwkAddr; // dirección Own NWK
static AppMessageBuffer_t appMessageBuffer; // mensajes de bus de direcciones
static uint8_t messageIdTx = 0; // transmission de identificador
de aplicación
static uint8_t messageIdRx = 0; // próximo identificador de
mensaje a recibir
static uint16_t actualDataLength = 0; // largo actual de trama de datos
a transmitir via red

// indicador de datos FIFO con variables relacionadas
static uint8_t apsDataIndFifo[APP_DATA_IND_BUFFER_SIZE];
static uint16_t apsDataIndFifoStart = 0;
static uint16_t apsDataIndFifoEnd = 0;

// Variables relacionadas con USART
static HAL_UsartDescriptor_t appUsartDescriptor; // USART descriptor
(requerido por stack)
static bool usartTxBusy = false; // USART estado de
transmisión
static uint8_t usartTxBuffer[APP_USART_TX_BUFFER_SIZE]; // USART Tx buffer
static uint8_t usartRxBuffer[APP_USART_RX_BUFFER_SIZE]; // USART Rx buffer

// parametros relacionados con la aplicación
static AppState_t appState = APP_INITIAL_STATE; // estado de la aplicación
static AppDataTransmissionState_t appDataTransmissionState;
static ZDO_StartNetworkReq_t networkParams; // pedido de parametros para
ZDO_StartNetworkReq
static APS_DataReq_t apsDataReq;

// parametros de finalización
static SimpleDescriptor_t simpleDescriptor = { APP_ENDPOINT, APP_PROFILE_ID, 1,
1, 0, 0, NULL, 0, NULL };
static APS_RegisterEndpointReq_t endpointParams;
static bool noIndications = false;

// indicador de tiempo de red durante su utilización.
// Tiempo utilizado entre APS_DataConf y APS_DataReq.
static HAL_AppTimer_t delayTimer;

#if APP_DETECT_LINK_FAILURE == 1
static uint8_t retryCounter = 0; // cantidad de reintento de envío
de datos
// Elimina solicitud, usado para el ruteo entre redes con comunicaciones perdidas

```

```

static ZDO_ZdpReq_t leaveReq;
#endif // APP_DETECT_LINK_FAILURE

/*****
                Sección de declaración de funciones
*****/
static void APS_DataIndication(APS_DataInd_t* dataInd);
static void APS_DataConf(APS_DataConf_t* confInfo);
static void ZDO_StartNetworkConf(ZDO_StartNetworkConf_t* confirmInfo);
static void initNetwork(void);
static void startNetwork(void);
static void networkSendData(bool newTransmission);

static void startBlinkTimer(void);
static void startingNetworkTimerFired(void);
static void delayTimerFired(void);

static void initSerialInterface(void);
static void usartStartSending(void);
static void usartBytesReceived(uint16_t readBytesLen);
static void usartWriteConf(void);

static uint16_t fifoFreeSpace(void);
static void fifoWriteData(uint8_t *data, uint16_t size);
static uint16_t fifoReadData(uint8_t *data, uint16_t size);

static void buttonReleased(uint8_t buttons);

#if APP_DETECT_LINK_FAILURE == 1
static void leaveNetwork(void);
static void zdpLeaveResp(ZDO_ZdpResp_t *zdpResp);
#endif // APP_DETECT_LINK_FAILURE

/*****
                Sección de implementación
*****/
/*****//**
\control de tareas de apliciación.

\param none.
\return none.
*****/
void APL_TaskHandler(void)
{
    switch (appState)
    {
        case APP_INITIAL_STATE:
            // Estado inicial del nodo
            initSerialInterface(); // inicialización de usart
            BSP_OpenButtons(NULL, buttonReleased); // habilitación de pulsadores
            BSP_OpenLeds(); // habilitación de LEDs
            initNetwork();
            SYS_PostTask(APL_TASK_ID);
            break;

        case APP_NETWORK_JOINING_STATE:
            startBlinkTimer();
            startNetwork();
            break;

        case APP_NETWORK_LEAVING_STATE:
        case APP_NETWORK_JOINED_STATE:
        default:

```

```

        break;
    }
}

/*****
 \inicialización de parametros de red
 *****/
static void initNetwork(void)
{
    DeviceType_t deviceType;

    // NWK preconfiguración de red para configuración de servidor
    CS_ReadParameter(CS_NWK_ADDR_ID, &nwkAddr);

    CS_WriteParameter(CS_NWK_UNIQUE_ADDR_ID, &(bool){true});

    // Si nwkAddr es 0 entonces consta del nodo coordinador

    if (0 == nwkAddr)
    {
#ifdef _SECURITY_
        { //Seteo a unidad como coordinador
            ExtAddr_t extAddr;
            CS_ReadParameter(CS_APS_TRUST_CENTER_ADDRESS_ID, &extAddr);
            CS_WriteParameter(CS_UID_ID, &extAddr);
        }
#endif // _SECURITY_

        deviceType = DEVICE_TYPE_COORDINATOR;
    }
    else
    {
        deviceType = DEVICE_TYPE_ROUTER;
    }

    // Seteo el tipo de dispositivo para configurar valores de servidor.
    CS_WriteParameter(CS_DEVICE_TYPE_ID, &deviceType);

    // inicio de procedimiento
    appState = APP_NETWORK_JOINING_STATE;
}

/*****
 \param confirmInfo - confirmación de parametros.
 \return none.
 *****/
void ZDO_StartNetworkConf(ZDO_StartNetworkConf_t* confirmInfo)
{
    HAL_StopAppTimer(&delayTimer); // detengo intervalo temporal

    if (confirmInfo->status == ZDO_SUCCESS_STATUS)
    {
        appDataTransmissionState = APP_DATA_TRANSMISSION_READY_STATE;
        appState = APP_NETWORK_JOINED_STATE;
    }

#ifdef APP_DETECT_LINK_FAILURE == 1
    retryCounter = 0;
#endif // APP_DETECT_LINK_FAILURE
    actualDataLength = 0;
}

```

```

BSP_OnLed(APP_NETWORK_STATUS_LED);

// configure tiempo utilizado entre APS_DataConf y APS_DataReq.

// Configuración de finalización de aplicación
endpointParams.simpleDescriptor = &simpleDescriptor;
endpointParams.APS_DataInd = APS_DataIndication;
// Registro punto final
APS_RegisterEndpointReq(&endpointParams);
}
else
{
    SYS_PostTask(APL_TASK_ID);
}
}

/*****
 \Inicio de red
 *****/
static void startNetwork(void)
{
    networkParams.ZDO_StartNetworkConf = ZDO_StartNetworkConf;
    // inicio red
    ZDO_StartNetworkReq(&networkParams);
}

#if APP_DETECT_LINK_FAILURE == 1
/*****
 \licencia de red.
 *****/
static void leaveNetwork(void)
{
    ZDO_MgmtLeaveReq_t *zdpLeaveReq = &leaveReq.req.reqPayload.mgmtLeaveReq;
    APS_UnregisterEndpointReq_t unregEndpoint;

    appState = APP_NETWORK_LEAVING_STATE;

    unregEndpoint.endpoint = endpointParams.simpleDescriptor->endpoint;
    APS_UnregisterEndpointReq(&unregEndpoint);

    leaveReq.ZDO_ZdpResp = zdpLeaveResp;
    leaveReq.reqCluster = MGMT_LEAVE_CLID;
    leaveReq.dstAddrMode = EXT_ADDR_MODE;
    leaveReq.dstExtAddr = 0;
    zdpLeaveReq->deviceAddr = 0;
    zdpLeaveReq->rejoin = 0;
    zdpLeaveReq->removeChildren = 1;
    zdpLeaveReq->reserved = 0;
    ZDO_ZdpReq(&leaveReq);
}

/*****
 \Abandono de respuesta de red
 *****/
static void zdpLeaveResp(ZDO_ZdpResp_t *zdpResp)
{
    BSP_OffLed(APP_NETWORK_STATUS_LED);

    // Try to rejoin the network
    appState = APP_NETWORK_JOINING_STATE;
    SYS_PostTask(APL_TASK_ID);
}

```

```

    (void)zdpResp;
}
#endif // APP_DETECT_LINK_FAILURE

/*****
 \inicio timer basado en HAL_AppTimer_t.
 *****/
static void startBlinkTimer(void)
{
    delayTimer.interval = APP_JOINING_INDICATION_PERIOD;
    delayTimer.mode     = TIMER_REPEAT_MODE;
    delayTimer.callback = startingNetworkTimerFired;
    HAL_StartAppTimer(&delayTimer);
}

/*****
 \función devolución timer.
 *****/
static void startingNetworkTimerFired(void)
{
    BSP_ToggleLed(APP_NETWORK_STATUS_LED);
}

/*****
 \Actualización de controlador de estado de eventos.
 *****/
void ZDO_MgmtNwkUpdateNotf(ZDO_MgmtNwkUpdateNotf_t *nwkParams)
{
    switch (nwkParams->status)
    {
        case ZDO_NETWORK_STARTED_STATUS:
            break;

        case ZDO_NETWORK_LOST_STATUS:
            {
                APS_UnregisterEndpointReq_t unregEndpoint;

                unregEndpoint.endpoint = endpointParams.simpleDescriptor->endpoint;
                APS_UnregisterEndpointReq(&unregEndpoint);

                BSP_OffLed(APP_NETWORK_STATUS_LED);

                // tratar de reincorporarse a la red
                appState = APP_NETWORK_JOINING_STATE;
                SYS_PostTask(APL_TASK_ID);
                break;
            }

        case ZDO_NWK_UPDATE_STATUS:
            break;

        default:
            break;
    }
}

/*****
 \breve activación de controlador de eventos.
 *****/
void ZDO_WakeUpInd(void)
{

```



```

}

/*****
 \envío de datos de red.
 *****/
static void networkSendData(bool newTransmission)
{
    if (APP_DATA_TRANSMISSION_SENDING_STATE == appDataTransmissionState)
    {
        appDataTransmissionState = APP_DATA_TRANSMISSION_BUSY_STATE;
        // indicamos que estamos enviando
        BSP_OnLed(APP_SENDING_STATUS_LED);

        if (newTransmission)
        {
            // se envía Nuevo mensaje de aplicación.

            appMessageBuffer.message.messageId = messageIdTx++;
            // Preparación de datos de aplicación
            apsDataReq.dstAddrMode = APS_SHORT_ADDRESS;           // corto

            // Dirección destino
            apsDataReq.dstAddress.shortAddress = nwkAddr ^ 1;
            apsDataReq.profileId = simpleDescriptor.AppProfileId; // perfil ID
            apsDataReq.dstEndpoint = simpleDescriptor.endpoint;  // Destino final
            apsDataReq.clusterId = APP_CLUSTER_ID;               // Cluster de
Destino
            apsDataReq.srcEndpoint = simpleDescriptor.endpoint;  // Fuente final
            apsDataReq.asdu = (uint8_t*) &appMessageBuffer.message; // Mensaje de
aplicación

            // Longitud de mensaje de aplicación
            apsDataReq.asduLength = actualDataLength +
sizeof(appMessageBuffer.message.messageId);
            apsDataReq.txOptions.acknowledgedTransmission = 1;    // Acknowledged
Transmisión habilitada
#ifdef APP_FRAGMENTATION
            apsDataReq.txOptions.fragmentationPermitted = 1;
#else
            apsDataReq.txOptions.fragmentationPermitted = 0;
#endif // APP_FRAGMENTATION
            apsDataReq.radius = 0;
            apsDataReq.APS_DataConf = APS_DataConf;               //confirmación de
control
        }

        APS_DataReq(&apsDataReq);
    }
}

/*****
 \confirmación de envío de datos de aplicación
 *****/
void APS_DataConf(APS_DataConf_t* confInfo)
{
    BSP_OffLed(APP_SENDING_STATUS_LED);

    if (APS_SUCCESS_STATUS != confInfo->status)
    {
#ifdef APP_DETECT_LINK_FAILURE == 1
        retryCounter++;
        if (MAX_RETRIES_BEFORE_REJOIN == retryCounter)

```

```

    {
        leaveNetwork();
    }
    else
#endif // APP_DETECT_LINK_FAILURE
    {
        // Datos no recibidos por punto final, reenviando.
        appDataTransmissionState = APP_DATA_TRANSMISSION_SENDING_STATE;
        networkSendData(false);
    }
    return;
}

#if APP_DETECT_LINK_FAILURE == 1
    retryCounter = 0;
#endif // Detección de falla de aplicación
    actualDataLength = 0;

#if APP_TRANSMISSION_DELAY > 0
    appDataTransmissionState = APP_DATA_TRANSMISSION_STOP_STATE;

    delayTimer.interval = APP_TRANSMISSION_DELAY;
    delayTimer.mode      = TIMER_ONE_SHOT_MODE;
    delayTimer.callback = delayTimerFired;
    HAL_StartAppTimer(&delayTimer);
#else
    appDataTransmissionState = APP_DATA_TRANSMISSION_READY_STATE;
    usartBytesReceived(0);
#endif
}

/*****
 \llamado de función.
 *****/
static void delayTimerFired(void)
{
    if (APP_DATA_TRANSMISSION_STOP_STATE == appDataTransmissionState)
        appDataTransmissionState = APP_DATA_TRANSMISSION_READY_STATE;
    else
        appDataTransmissionState = APP_DATA_TRANSMISSION_SENDING_STATE;
    usartBytesReceived(0);
}

/*****
 \Indicador de datos de controlador de aplicación.
 *****/
void APS_DataIndication(APS_DataInd_t* indData)
{
    AppMessage_t *appMessage = (AppMessage_t *) indData->asdu;

    // Indicador de datos recibidos
    BSP_OnLed(APP_RECEIVING_STATUS_LED);

    if (appMessage->messageId == messageIdRx)
    {
        fifoWriteData(appMessage->data, indData->asduLength - 1);

        usartStartSending();

        if (fifoFreeSpace() < APP_APS_PAYLOAD_SIZE)
        {
            APS_StopEndpointIndication(APP_ENDPOINT); // Disable APS Data Indications

```

```

        noIndications = true;
    }
}

messageIdx = appMessage->messageId + 1;

BSP_OffLed(APP_RECEIVING_STATUS_LED);
}
/*****
 \Inicialización de USART o VCP (depende del seteo).
 *****/
static void initSerialInterface(void)
{
    usartTxBusy = false;

    appUsartDescriptor.tty           = USART_CHANNEL;
    appUsartDescriptor.mode          = USART_MODE_ASYNC;
    appUsartDescriptor.baudrate      = USART_BAUDRATE_38400;
    appUsartDescriptor.dataLength    = USART_DATA8;
    appUsartDescriptor.parity        = USART_PARITY_NONE;
    appUsartDescriptor.stopbits      = USART_STOPBIT_1;
    appUsartDescriptor.rxBuffer      = usartRxBuffer;
    appUsartDescriptor.rxBufferLength = sizeof(usartRxBuffer);
    appUsartDescriptor.txBuffer       = NULL; // use callback mode
    appUsartDescriptor.txBufferLength = 0;
    appUsartDescriptor.rxCallback     = usartBytesReceived;
    appUsartDescriptor.txCallback     = usartWriteConf;
    appUsartDescriptor.flowControl    = USART_FLOW_CONTROL_HARDWARE;

#if BSP_ENABLE_RS232_CONTROL == 1
    BSP_EnableRs232();
#endif // BSP_ENABLE_RS232_CONTROL

    OPEN_USART(&appUsartDescriptor);
}

/*****
 Inicialización de envoi de datos por Puerto serie.
 *****/
static void usartStartSending(void)
{
    uint16_t size;

    if (!usartTxBusy)
    {
        size = fifoReadData(usartTxBuffer, APP_USART_TX_BUFFER_SIZE);
        WRITE_USART(&appUsartDescriptor, usartTxBuffer, size);

        usartTxBusy = true;

        if (noIndications && (fifoFreeSpace() >= APP_APS_PAYLOAD_SIZE))
        {
            APS_ResumeEndpointIndication(APP_ENDPOINT); // Enable APS Data Indications
            noIndications = false;
        }
    }
}

/*****
 \Confirmación de dato recibido, y configuración para nuevo envío
 *****/

```

```

static void usartWriteConf(void)
{
    usartTxBusy = false;

    if (APP_DATA_IND_BUFFER_SIZE != fifoFreeSpace())
        usartStartSending();
}

/*****
 \Nuevo dato recibido por Puerto serie.
 *****/
static void usartBytesReceived(uint16_t bytesToRead)
{
    if (APP_NETWORK_JOINED_STATE != appState)
        return;
    if ((APP_DATA_TRANSMISSION_BUSY_STATE == appDataTransmissionState)
        || (APP_DATA_TRANSMISSION_STOP_STATE == appDataTransmissionState))
        return;

    if (actualDataLength < APP_MAX_PACKET_SIZE)
    {
        actualDataLength += (uint8_t) READ_USART(&appUsartDescriptor,
            appMessageBuffer.message.data + actualDataLength,
            APP_MAX_PACKET_SIZE - actualDataLength);
    }

    #if APP_DELAY_BEFORE_SEND > 0
    if (actualDataLength < APP_MAX_PACKET_SIZE)
    {
        if ((APP_DATA_TRANSMISSION_READY_STATE == appDataTransmissionState) &&
            (actualDataLength > 0))
        {
            appDataTransmissionState = APP_DATA_TRANSMISSION_WAIT_STATE;
            delayTimer.interval = APP_DELAY_BEFORE_SEND;
            delayTimer.mode = TIMER_ONE_SHOT_MODE;
            delayTimer.callback = delayTimerFired;
            HAL_StartAppTimer(&delayTimer);
            return;
        }
        if (APP_DATA_TRANSMISSION_WAIT_STATE == appDataTransmissionState)
            return;
    }
    else if (APP_DATA_TRANSMISSION_WAIT_STATE == appDataTransmissionState)
        HAL_StopAppTimer(&delayTimer);
    #endif

    if (actualDataLength > 0)
    {
        appDataTransmissionState = APP_DATA_TRANSMISSION_SENDING_STATE;
        networkSendData(true);
    }

    (void)bytesToRead;
}

/*****
 *****/
static uint16_t fifoFreeSpace(void)
{
    int16_t free = apsDataIndFifoStart - apsDataIndFifoEnd;
}

```

```

if (apsDataIndFifoStart <= apsDataIndFifoEnd)
    free += sizeof(apsDataIndFifo);

return free;
}

/*****
*****
static void fifoWriteData(uint8_t *data, uint16_t size)
{
    uint16_t i;

    for (i = 0; i < size; i++)
    {
        apsDataIndFifo[apsDataIndFifoEnd++] = data[i];
        if (apsDataIndFifoEnd == sizeof(apsDataIndFifo))
            apsDataIndFifoEnd = 0;
    }
}

/*****
\Lectura de indicador de datos
*****
static uint16_t fifoReadData(uint8_t *data, uint16_t size)
{
    uint16_t read = 0;
    uint16_t i;

    for (i = 0; i < size; i++)
    {
        if (apsDataIndFifoStart == apsDataIndFifoEnd)
            break;

        data[i] = apsDataIndFifo[apsDataIndFifoStart++];
        read++;

        if (apsDataIndFifoStart == sizeof(apsDataIndFifo))
            apsDataIndFifoStart = 0;
    }

    return read;
}

/*****
\Control de evento de boton
*****
static void buttonReleased(uint8_t aButton)
{
    switch (aButton)
    {
        case BSP_KEY0: // SW1 pulsado
            if ((APP_NETWORK_JOINED_STATE == appState) &&
                (APP_DATA_TRANSMISSION_READY_STATE == appDataTransmissionState))
            {
                memset(&appMessageBuffer.message.data, 0xdd, APP_MAX_PACKET_SIZE);
                actualDataLength = APP_MAX_PACKET_SIZE;
                appDataTransmissionState = APP_DATA_TRANSMISSION_SENDING_STATE;
                networkSendData(true);
                break;
            }
    }
}

```

```

        default:
            break;
    }
}
#endif _BINDING_
/*****
*****/
void ZDO_BindIndication(ZDO_BindInd_t *bindInd)
{
    (void)bindInd;
}

/*****
*****/
void ZDO_UnbindIndication(ZDO_UnbindInd_t *unbindInd)
{
    (void)unbindInd;
}
#endif //_BINDING_

/*****//**
 \Inicio Principal del programa
 *****/
int main(void)
{
    SYS_SysInit();

    for(;;)
    {
        SYS_RunTask();
    }
}
// eof peer2Peer.c

```

### 3.5- WSN Demo

A continuación se presenta el código de la aplicación, el cual ha sido documentado en castellano para su mejor interpretación. El mismo pertenece al stack Lightweight Mesh, Release Version: 1.1.1 de fecha Agosto 15, 2013, el ejemplo WSNDemo.

```

/*
    BREVE DESCRIPCIÓN DE LA APLICACIÓN:

```

```

En forma básica, esta aplicación genera una red re-transmitiendo los valores generados por funciones random, en las variables de estructuras, "appMsg.sensors.battery", "appMsg.sensors.temperature", y "appMsg.sensors.light". Dicha aplicación es utilizable para la transmisión de datos de sensores por medio de la modificación de las variables definidas en las estructura "sensors", y la generación de datos, es decir, reemplazando las funciones random por la adquisición de datos correspondiente, ya sea SPI, TWI, UART, entre otros.

```

```

Dentro de la aplicación WSN se puede cambiar la máscara del canal de red, los tiempos de espera de nodo y se puede restablecer cualquier nodo de forma remota.

```

```

El procedimiento para cargar este programa es el mismo visto anteriormente. Se abre el archivo WSNDemoApp.aps que se encuentra en la carpeta de programas AVR. Se compila y a continuación se programan todos los dispositivos con la misma imagen. Una vez hecho esto hay que configurar los interruptores de cada dispositivo ya sea coordinador, router, o dispositivo final. */

```

```

/*- Includes -----*/
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "config.h"
#include "hal.h"
#include "phy.h"
#include "sys.h"
#include "nwk.h"
#include "nwkRoute.h"
#include "nwkSecurity.h"
#include "sysTimer.h"
#include "halUart.h"
#include "halSleep.h"
#include "halled.h"

#ifdef APP_ENABLE_OTA
#include "otaClient.h"
#endif

/*- Definitions -----*/
#if APP_ADDR == 0
#define APP_CAPTION      "Coordinator"
#define APP_NODE_TYPE   0
#define APP_COORDINATOR 1
#elif APP_ADDR < 0x8000
#define APP_CAPTION      "Router"
#define APP_NODE_TYPE   1
#define APP_ROUTER      1
#else
#define APP_CAPTION      "End Device"
#define APP_NODE_TYPE   2
#define APP_ENDDEVICE   1
#endif

#define APP_CAPTION_SIZE (sizeof(APP_CAPTION) - 1)

#define LED_NETWORK      0
#define LED_DATA         1

/*- Types -----*/
typedef struct PACK AppMessage_t
{
    uint8_t    messageType;
    uint8_t    nodeType;
    uint64_t   extAddr;
    uint16_t   shortAddr;
    uint32_t   softVersion;
    uint32_t   channelMask;
    uint16_t   panId;
    uint8_t    workingChannel;
    uint16_t   parentShortAddr;
    uint8_t    lqi;
    int8_t     rssi;

    struct PACK
    {
        uint8_t    type;
        uint8_t    size;
        int32_t    battery;
        int32_t    temperature;
        int32_t    light;
    };
};

```

```

    } sensors;

    struct PACK
    {
        uint8_t    type;
        uint8_t    size;
        char       text[APP_CAPTION_SIZE];
    } caption;
} AppMessage_t;

typedef enum AppState_t
{
    APP_STATE_INITIAL,
    APP_STATE_SEND,
    APP_STATE_WAIT_CONF,
    APP_STATE_SENDING_DONE,
    APP_STATE_WAIT_SEND_TIMER,
    APP_STATE_PREPARE_TO_SLEEP,
    APP_STATE_SLEEP,
    APP_STATE_WAKEUP,
} AppState_t;

/*- Variables -----*/
static AppState_t appState = APP_STATE_INITIAL;

#if APP_ROUTER || APP_ENDDEVICE
static NWK_DataReq_t nwkDataReq;
static SYS_Timer_t appNetworkStatusTimer;
static bool appNetworkStatus;
#endif

static AppMessage_t appMsg;
static SYS_Timer_t appDataSendingTimer;

/*- Implementations -----*/

/*****
*****
void HAL_UartBytesReceived(uint16_t bytes)
{
    for (uint16_t i = 0; i < bytes; i++)
        HAL_UartReadByte();
    HAL_LedToggle(2);
}

/*****
*****
static void appSendMessage(uint8_t *data, uint8_t size)
{
    uint8_t cs = 0;

    HAL_UartWriteByte(0x10);
    HAL_UartWriteByte(0x02);

    for (uint8_t i = 0; i < size; i++)
    {
        if (data[i] == 0x10)
        {
            HAL_UartWriteByte(0x10);
            cs += 0x10;
        }
        HAL_UartWriteByte(data[i]);
    }
}

```



```

    cs += data[i];
}

HAL_UartWriteByte(0x10);
HAL_UartWriteByte(0x03);
cs += 0x10 + 0x02 + 0x10 + 0x03;

HAL_UartWriteByte(cs);
}

/*****
*****
static bool appDataInd(NWK_DataInd_t *ind)
{
    AppMessage_t *msg = (AppMessage_t *)ind->data;

    HAL_LedToggle(LED_DATA);

    msg->lqi = ind->lqi;
    msg->rssi = ind->rssi;

    appSendMessage(ind->data, ind->size);
    return true;
}

/*****
*****
static void appDataSendingTimerHandler(SYS_Timer_t *timer)
{
    if (APP_STATE_WAIT_SEND_TIMER == appState)
        appState = APP_STATE_SEND;
    else
        SYS_TimerStart(&appDataSendingTimer);

    (void)timer;
}

#if APP_ROUTER || APP_ENDDEVICE
/*****
*****
static void appNetworkStatusTimerHandler(SYS_Timer_t *timer)
{
    HAL_LedToggle(LED_NETWORK);
    (void)timer;
}
#endif

/*****
*****
#if APP_ROUTER || APP_ENDDEVICE
static void appDataConf(NWK_DataReq_t *req)
{
    HAL_LedOff(LED_DATA);

    if (NWK_SUCCESS_STATUS == req->status)
    {
        if (!appNetworkStatus)
        {
            HAL_LedOn(LED_NETWORK);
            SYS_TimerStop(&appNetworkStatusTimer);
            appNetworkStatus = true;
        }
    }
}
#endif

```

```

    }
    else
    {
        if (appNetworkStatus)
        {
            HAL_LedOff(LED_NETWORK);
            SYS_TimerStart(&appNetworkStatusTimer);
            appNetworkStatus = false;
        }
    }

    appState = APP_STATE_SENDING_DONE;
}
#endif

/*****
*****
static void appSendData(void)
{
#ifdef NWK_ENABLE_ROUTING
    appMsg.parentShortAddr = NWK_RouteNextHop(0, 0);
#else
    appMsg.parentShortAddr = 0;
#endif

    appMsg.sensors.battery      = rand() & 0xffff;
    appMsg.sensors.temperature = rand() & 0x7f;
    appMsg.sensors.light       = rand() & 0xff;

#ifdef APP_COORDINATOR
    appSendMessage((uint8_t *)&appMsg, sizeof(appMsg));
    SYS_TimerStart(&appDataSendingTimer);
    appState = APP_STATE_WAIT_SEND_TIMER;
#else
    nwkDataReq.dstAddr = 0;
    nwkDataReq.dstEndpoint = APP_ENDPOINT;
    nwkDataReq.srcEndpoint = APP_ENDPOINT;
    nwkDataReq.options = NWK_OPT_ACK_REQUEST | NWK_OPT_ENABLE_SECURITY;
    nwkDataReq.data = (uint8_t *)&appMsg;
    nwkDataReq.size = sizeof(appMsg);
    nwkDataReq.confirm = appDataConf;

    HAL_LedOn(LED_DATA);
    NWK_DataReq(&nwkDataReq);

    appState = APP_STATE_WAIT_CONF;
#endif
}

#ifdef PHY_ENABLE_RANDOM_NUMBER_GENERATOR
/*****
*****
void PHY_RandomConf(uint16_t rnd)
{
    srand(rnd);
}
#endif

/*****
*****
static void appInit(void)
{

```

```

appMsg.messageType          = 1;
appMsg.nodeType             = APP_NODE_TYPE;
appMsg.extAddr              = APP_ADDR;
appMsg.shortAddr            = APP_ADDR;
appMsg.softVersion         = 0x01010100;
appMsg.channelMask         = (1L << APP_CHANNEL);
appMsg.panId                = APP_PANID;
appMsg.workingChannel       = APP_CHANNEL;
appMsg.parentShortAddr     = 0;
appMsg.lqi                  = 0;
appMsg.rssi                 = 0;

appMsg.sensors.type        = 1;
appMsg.sensors.size        = sizeof(int32_t) * 3;
appMsg.sensors.battery     = 0;
appMsg.sensors.temperature = 0;
appMsg.sensors.light       = 0;

appMsg.caption.type        = 32;
appMsg.caption.size        = APP_CAPTION_SIZE;
memcpy(appMsg.caption.text, APP_CAPTION, APP_CAPTION_SIZE);

#if APP_COORDINATOR
// Enable RCB_BB RS232 level converter
#if defined(PLATFORM_RCB128RFA1) || defined(PLATFORM_RCB256RFR2)
    DDRD = (1 << 4) | (1 << 6) | (1 << 7);
    PORTD = (0 << 4) | (1 << 6) | (1 << 7);
#endif

#if defined(PLATFORM_RCB231)
    DDRC = (1 << 4) | (1 << 6) | (1 << 7);
    PORTC = (0 << 4) | (1 << 6) | (1 << 7);
#endif
#endif

#if defined(PLATFORM_XPLAINED_PRO) && defined(HAL_ATMEGA256RFR2)
// Enable chip antenna
DDRG = (1 << 1);
PORTG = (0 << 1);

DDRF = (1 << 2);
PORTF = (1 << 2);
#endif

HAL_LedInit();

NWK_SetAddr(APP_ADDR);
NWK_SetPanId(APP_PANID);
PHY_SetChannel(APP_CHANNEL);
#ifdef PHY_AT86RF212
    PHY_SetBand(APP_BAND);
    PHY_SetModulation(APP_MODULATION);
#endif
PHY_SetRxState(true);

#ifdef NWK_ENABLE_SECURITY
    NWK_SetSecurityKey((uint8_t *)APP_SECURITY_KEY);
#endif

NWK_OpenEndpoint(APP_ENDPOINT, appDataInd);

appDataSendingTimer.interval = APP_SENDING_INTERVAL;

```

```

appDataSendingTimer.mode = SYS_TIMER_INTERVAL_MODE;
appDataSendingTimer.handler = appDataSendingTimerHandler;

#if APP_ROUTER || APP_ENDDEVICE
appNetworkStatus = false;
appNetworkStatusTimer.interval = 500;
appNetworkStatusTimer.mode = SYS_TIMER_PERIODIC_MODE;
appNetworkStatusTimer.handler = appNetworkStatusTimerHandler;
SYS_TimerStart(&appNetworkStatusTimer);
#else
HAL_LedOn(LED_NETWORK);
#endif

#ifdef PHY_ENABLE_RANDOM_NUMBER_GENERATOR
PHY_RandomReq();
#endif

appState = APP_STATE_SEND;
}

/*****
*****
static void APP_TaskHandler(void)
{
switch (appState)
{
case APP_STATE_INITIAL:
{
appInit();
} break;

case APP_STATE_SEND:
{
appSendData();
} break;

case APP_STATE_SENDING_DONE:
{
#if APP_ENDDEVICE
appState = APP_STATE_PREPARE_TO_SLEEP;
#else
SYS_TimerStart(&appDataSendingTimer);
appState = APP_STATE_WAIT_SEND_TIMER;
#endif
} break;

case APP_STATE_PREPARE_TO_SLEEP:
{
if (!NWK_Busy())
{
NWK_SleepReq();
appState = APP_STATE_SLEEP;
}
} break;

case APP_STATE_SLEEP:
{
HAL_LedClose();
HAL_Sleep(APP_SENDING_INTERVAL);
appState = APP_STATE_WAKEUP;
} break;
}
}

```

```

    case APP_STATE_WAKEUP:
    {
        NWK_WakeupReq();

        HAL_LedInit();
        HAL_LedOn(LED_NETWORK);

        appState = APP_STATE_SEND;
    } break;

    default:
        break;
}
}

/*****
*****
int main(void)
{
    SYS_Init();
    HAL_UartInit(38400);
#ifdef APP_ENABLE_OTA
    OTA_ClientInit();
#endif

    while (1)
    {
        SYS_TaskHandler();
        HAL_UartTaskHandler();
#ifdef APP_ENABLE_OTA
        OTA_ClientTaskHandler();
#endif
        APP_TaskHandler();
    }
}

```

### 3.6- Contiki – 6LowPan

Tal como se comento en la sección 3.1.4 no se ha profundizado en Contiki, ya que implica el estudio de programación de sistemas embebidos utilizando un sistema operativo de tiempo real (RTOS). Lo cual no es el objetivo en esta etapa del proyecto.

### 3.7- Conclusión.

Estudiados y ensayados los distintos stack y aplicaciones de ejemplo disponibles por parte de ATMEL para la plataforma seleccionada se concluye que para la puesta en marcha de la microestación es suficiente el ejemplo de P2P del stack de lightmesh, mientras que para tomar como base para la implementación de la red se debe usar el WSDemo del stack BitCloud, el cual permite la formación de una red tipo mesh y la participación de múltiples dispositivos.

En lo que respecta a la aplicación propiamente dicha, se decidió suspenderla, ya que para efectuar un desarrollo con posibilidad de transferencia debía efectuarse sobre una plataforma que no se encuentre discontinuada. Sin embargo el estudio, los ensayos y pruebas realizadas aportaron un gran comprensión del tema, lo que permitirá encarar una nueva plataforma más fácilmente.

## 4- Estación colectora

### 4.1- Descripción General – Hardware

Para el desarrollo de la estación colectora se utilizó un kit de evaluación de Atmel SAM3X-EK [78] el cual puede verse en la figura N°4.1, que cuenta con un micro ARM CORTEX M3 SAM3X8H [79] y la mayoría de las interfaces disponibles. La selección de esta plataforma se justifica en que la misma posee la capacidad y los periféricos necesarios para el proyecto, además fue la empleada en el proyecto Prince C130, predecesor del actual proyecto.

Para la comunicación a través de la red celular se utilizó un módulo de la empresa Telit, el GL865 QUAD [80], figura N°4.2, en este caso no se continuó con el módulo empleado en el proyecto Prince C130 debido a que el mismo fue discontinuado.



Figura N°4.1- Kit Atmel ATSAM3X-EK

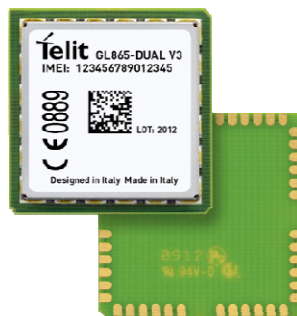


Figura 4.2 - Módulo GSM/GPRS - Kit 2G con chip Telit GL-865

### 4.2- Firmware

La estación colectora es la encargada de recibir los datos de la microestación por puerto serie (UART0), procesarlos, validarlos, completarlos y retransmitirlos a través de la interfaz Ethernet o a través del módulo de GSM/GPRS conectado al puerto serie (UART1). Un puerto serie extra se utiliza como consola de debug. La alimentación del módulo GSM/GPRS se maneja a través de un relay con la estación colectora la cual puede apagarlo y encenderlo en cualquier momento.

La señalización de los estados del equipo se realizan con 2 leds del kit de Atmel. El led azul titilando indica que esta intentando conectarse por GPRS a la red del proveedor y tratando de obtener una ip del mismo para lograr comunicarnos por TCP/IP. Si este permanece fijo es que ya tenemos la ip y la conexión con el proveedor. El led ambar titilando indica que esta intentando conectarse por TCP/IP a la ip del servidor remoto. Si este permanece fijo es que ya logramos la conexión con el servidor remoto y se puede empezar a enviar mensajes y/o datos al servidor.

#### 4.2.1- Descripción general

La figura 4.3 presenta el diagrama de flujo de la aplicación implementada.

El programa comienza con la inicialización del microcontrolador (clock, pll) y los periféricos a utilizar como los puertos serie (UART), Ethernet, y las entradas salidas a controlar. Luego se definen las tareas, una asociada al envío de datos por Ethernet y otra al envío de datos por GPRS, con sus prioridades y sus buffers, y se inician junto con la habilitación de las interrupciones.

La tarea de Ethernet se dispara primero e intentar obtener ip por DHCP, si se logra se conecta con el servidor a una ip previamente definida y verifica si se tiene conexión real con el mismo. Si todo está en orden queda a la espera de un comando de ACK.

Si dicho comando es recibido y se han recibido datos por el puerto serie, provenientes de las microestaciones, estos son enviados al servidor. Si los datos son enviados satisfactoriamente son marcados en la flash como enviados para no volver a enviarlos.

El ciclo de Ethernet de obtención de ip lo realiza 3 veces y si no es posible dispara la tarea de GPRS. En la misma enciende el modem, obtiene ip del proveedor e intenta conectarse al servidor, luego chequea conexión y si tiene datos a enviar (ingresados también por la interrupción RS232) los envía inmediatamente y los marca en la flash como enviados. Este ciclo también lo realiza 3 veces la obtención de ip del proveedor, si no se logra se pasa a desconectar el modem y disparar la tarea de Ethernet. Y así sucesivamente intercambiando las tareas y buscando el medio para el envío de los datos.

Paralelamente a la ejecución de estas tareas esta activada la interrupción del puerto serie. Los datos recibidos son chequeados que sean correctos en su estructura de protocolo, y en caso afirmativo son almacenados en memoria para que sean transmitidos al servidor cuando haya un canal disponible.

#### 4.2.2- Herramientas y entorno de trabajo

Para la realización del proyecto se utilizo el software **ATMEL STUDIO 6.1.2665 Service pack 1** y el **Atmel Software Framework 3.9.1.780**. Si bien no es la ultima version se trabajo con esta para evitar problemas de migración con el firmware desarrollado en el proyecto PROINCE C130. Se partió del ejemplo **FreeRTOS with lwIP Example – SAM3X-EK** que utiliza las librerías de LWIP para Ethernet y ejemplos de manejo de interrupciones y UARTs.

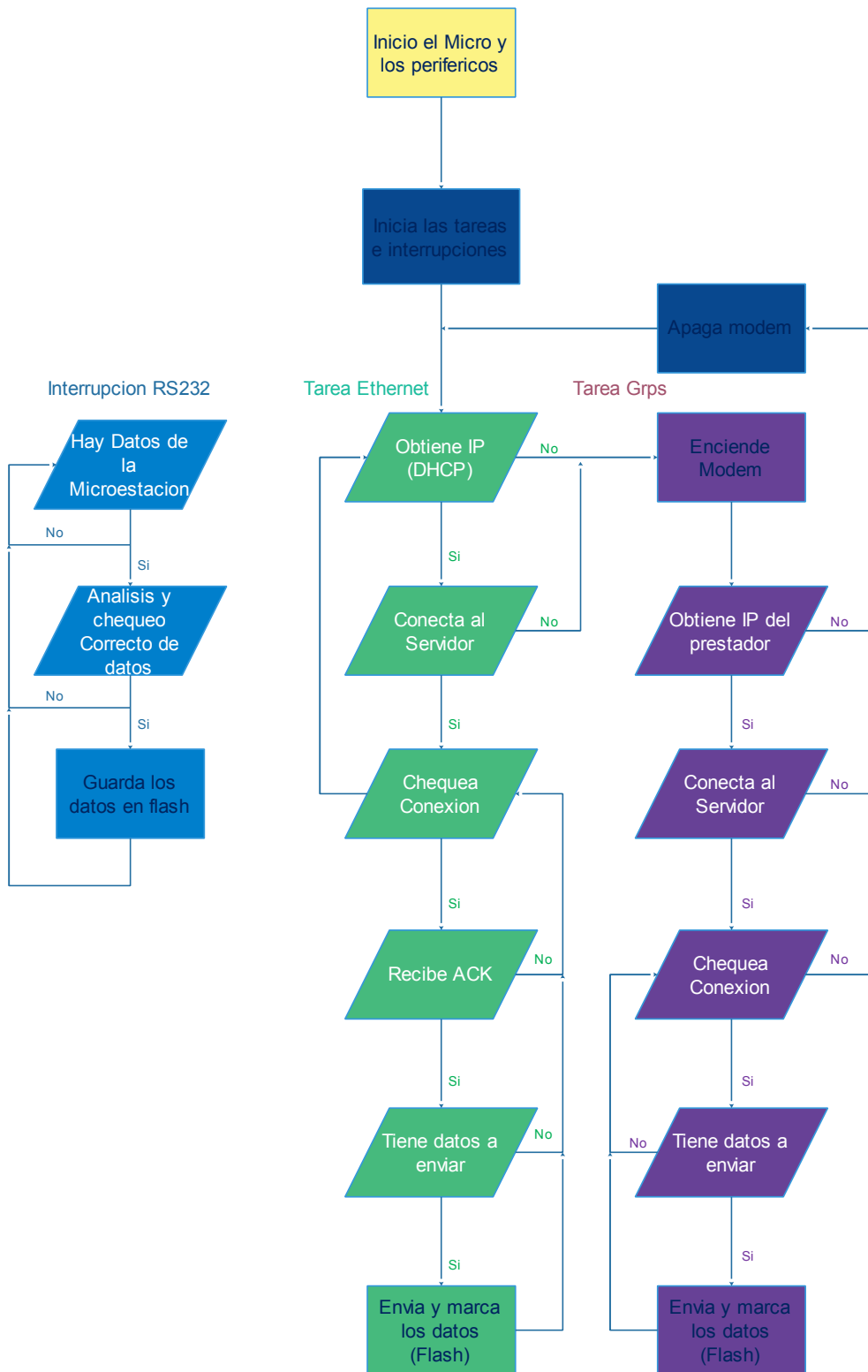


Figura 4.3 – Diagrama de flujo.

#### 4.2.3- Proyecto Base

Para el desarrollo del firmware se tomo como base los desarrollos realizados en el proyecto “Prounce C130 Sistema inalámbrico de microsensores para aplicaciones agropecuarias”. En dicho desarrollo se habían planteado las siguientes etapas:



- Recepción de los datos a través del puerto serie
- Transmisión de datos sobre Ethernet a un socket TCP/UDP
- Transmisión de datos sobre Ethernet a un servidor de correo electrónico.
- Transmisión de datos sobre GPRS a un socket TCP/UDP
- Transmisión de datos sobre GPRS a un servidor de correo electrónico.

De las mismas las tres primeras se llegaron a implementar encontrándose algunos problemas y se estaba en desarrollo las últimas dos. Como conclusiones de dicho trabajo se plantearon las siguientes mejoras:

- Se debe contar con un buffer de 10 mensajes en el kit tanto para el puerto esclavo como para los mensajes hacia el Server y en anillo.
- Poder configurar por el puerto esclavo cosas como IP del Server, puerto, email o TCP o UDP de conexión, etc.
- Reconocimiento automatico de la IP asignada.
- Analizar el stack TCP/IP de forma que no sea necesario el envío de un paquete desde el server para recibir los datos del cliente.
- Analizar el stack SMTP de forma que se pueda utilizar cuentas con seguridad SSL.

#### **4.2.4- Nuevo planteo**

Sobre la base del proyecto mencionado en la sección anterior se plantean las siguientes acciones:

- Migrar el código desarrollado para la comunicación GSM/GPRS realizado para un modulo Motorola G24 a un modulo Telit GL865.
- Generación de buffer de mensajes para almacenamiento de los mismos en caso de no tener conexión (Grabación de los datos en Flash).
- Configuración de los datos de conexión por puerto serie (IP, puerto, APN, etc).
- Envío de datos con seguridad SSL.
- Mejoramiento de las rutinas de obtención de IP por DHCP (detección de desconexión, políticas de reintento, reconocimiento automático de IP, etc).
- Analizar el stack TCP/IP de forma que no sea necesario el envío de un paquete desde el server para recibir los datos del cliente.
- Unificar en un solo código trabajando bajo el sistema operativo de tiempo real ambos métodos de transmisión (Ethernet y GPRS).

### **4.3 – Protocolo microestación.**

#### **4.3.1 –Detalle de la Trama de datos**

A continuación se detalla la trama de datos que utiliza el protocolo. La misma contempla el monitoreo de sensores y actuadores, se utiliza la coma (,) para separar campos de información. Todas las tramas empiezan con un encabezado \$D y terminan con #W

Ejemplo

```
$D,<estación>,<subestación>,<sen/act_1>,<sen/act_2>,<sen/act_3>,...,<sen/act_9>,<sen_act_10>,<CSUM>,#W
```

Donde:

\$D	Encabezado.
<estación>	Identificador de 8 bytes/caracteres. El mismo contendrá el número de serie de la estación (64 bits provenientes de un DS2401 o equivalente) o un nombre que lo identifique (si el nombre es menor de 8 caracteres, los faltantes serán completados con espacios).
<subestación>	Identificador de 1 carácter (0-9). En caso de que la microestación posea actuadores y/o más de 10 sensores, estos serán informados como pertenecientes a diferentes subestaciones. Valores de 0 a 4 representan sensores, valores de 5 a 9 actuadores.
<sen/act_n>	Valor numérico hexadecimal del sensor o actuador. 2 bytes (16 bits) Se usa el bit 15 (MSB) para indicar si esta presente el sensor o actuador (un 1 en este bit indica la presencia del sensor/actuador) y los restantes 15bits para el valor del sensor/actuador. Si los mismos son discretos se usa 0x8000 para OFF y 0xFFFF para ON.
<CSUM>	Checksum, código de verificación de la información (2 caracteres) que representan un número hexadecimal cuyo valor es el XOR lógico de todos los bytes/caracteres entre el \$ y el CSUM sin incluirlos (se incluye la coma previa al CSUM).
#W	Fin de trama.

Ejemplo real.

**\$D,ESTAC\_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,7D,#W**

**\$D,ESTAC\_01,5,8006,8032,8000,0000,F000,001C,FFFF,FFFF,FFFF,8000,9B,#W**

#### 4.3.2 –Detalle de la Trama de comandos

A continuación se detalla la trama de comandos que utiliza el protocolo, se utiliza la coma (,) para separar campos de información, todas las tramas empiezan con un encabezado: \$C y terminan con #W

Ejemplo

\$C,<estación>,<subestación>,<actuador1>,<actuador2>,<actuador3>,<actuador4>,<actuador5>,<actuador6>,<actuador7>,<actuador8>,<actuador9>,<actuador10>,<CSUM>,<#W

Donde:

\$C	Encabezado
<estación>	Identificador de 8 bytes/caracteres. El mismo contendrá el número de serie de la estación (64 bits provenientes de un DS2401 o equivalente) o un nombre que lo identifique (si el nombre es menor de 8 caracteres, los faltantes serán completados con espacios).
<subestación>	Identificador de 1 carácter (0-9). En caso de que la microestación posea actuadores y/o más de 10 sensores, estos serán informados como pertenecientes a diferentes

	subestaciones. Valores de 0 a 4 representan sensores, valores de 5 a 9 actuadores.
<actuador>	Valor numérico hexadecimal del actuador. 2 bytes (16 bits). Se usa el bit 15 (MSB) para indicar si está presente el actuador (un 1 en este bit indica la presencia del mismo) y los restantes 15bits para el valor. Si actuador es discreto se usa 0x8000 para OFF y 0xFFFF para ON.
<CSUM>	Checksum, código de verificación de la información (2 caracteres) que representan un número hexadecimal cuyo valor es el XOR lógico de todos los bytes/caracteres entre el \$ y el CSUM sin incluirlos (se incluye la coma previa al CSUM).
#W	Fin de trama

En caso de no tener todos los sensores o actuadores instalados, el valor de los 2 bytes será FFFF.

Ejemplo real

**\$C,ESTAC\_01,5,8032,8019,8000,0000,F000,F000,001C,FFFF,FFFF,FFFF,F3,#W**

#### 4.4- Desarrollo

El firmware se comenzó desde la base del ejemplo de FreeRTOS + LWIP (**FreeRTOS with lwIP Example – SAM3X-EK**). FreeRTOS es un sistema operativo de tiempo real utilizado en los microcontroladores actuales, mientras que LWIP es un stack TCP/IP libre, es decir un conjunto de librerías para el manejo de la comunicación por TCP/IP.

Los archivos modificados, son los que tenían alguna función o definición y se cambian por otra, y los agregados, son archivos en donde no había ninguna definición o función y se escribe una nueva sin interferir en su uso anterior porque no existía. Del proyecto original se modificaron 8 archivos y se agregaron 4 archivos más al proyecto, los cuales se detallan a continuación.

- **Archivos modificados.**  
*Sam3x\_ek.h , main.c , ethernet.c , opt.h , debug.h , conf\_uart\_serial.h , conf\_lwip\_threads.h, conf\_board.h.*
- **Archivos agregados.**  
*BasicSocket.c , BasicSOCKET.h , uart\_proc.c, init.c*

A raíz de que se realizó el firmware sobre un Sistema Operativo de Tiempo Real (RTOS), el flujo de programa se maneja con tareas las que describimos a continuación:

- IDLE: tarea de background.
- ETHLAUNCH: Encargado de manejar el driver de Ethernet y el protocolo TCP/IP
- WEBSERVER: a través de la cual un navegador puede acceder al kit, estación colectora, y monitorear las variable online.
- SOCKET: encargada de abrir el puerto y realizar la conexión al servidor a través de la interfaz Ethernet.
- GSM: idem SOCKET pero por GSM/GPRS

Las mismas se encuentran definidas en:

portTASK\_FUNCTION( vBasicSocketServer, pvParameters ) tarea SOCKET en BasicSOCKET.c  
portTASK\_FUNCTION(vStartEthernetTask, pvParameters) tarea ETHLAUNCH en ethernet.c  
portTASK\_FUNCTION( vBasicWEBServer, pvParameters ) tarea WEBSERVER en BasicWEB.c  
static void task\_gsm(void \*pvParameters) tarea GSM en main.c  
static portTASK\_FUNCTION( prvIdleTask, pvParameters ) tarea IDLE en tasks.c

File:Sam3x\_ek.h

Acción: Agregado

Detalle: Estas definiciones sirven para el manejo de la alimentación del modem GSM/GPRS y la configuración de la UART y la consola.

```

/*! Power GSM (power) */
#define POWER_GSM_GPIO      (PIO_PE0_IDX)
#define POWER_GSM_FLAGS    (PIO_TYPE_PIO_OUTPUT_0 | PIO_DEFAULT)
#define POWER_GSM_PIO      PIOE
#define PINS_USART0        (PIO_PA10A_RXD0 | PIO_PA11A_TXD0)
#define PINS_USART0_PORT   IOPORT_PIOA
#define PINS_USART0_MASK   (PIO_PA10A_RXD0 | PIO_PA11A_TXD0)
#define CONSOLE_USART1     USART1
#define CONSOLE_USART1_ID  ID_USART1

```

File:ethernet.c

Acción: Agregado.

Detalle: Si usa la definición de SOCKET\_USED se habilita la tarea del SOCKET para la comunicación por Ethernet.

```

#if (SOCKET_USED == 1)
#include "BasicSOCKET.h"
#endif

#if (SOCKET_USED == 1)
/* Create the SOCKET server task. This uses the lwIP RTOS abstraction layer. */
sys_thread_new("SOCK", vBasicSocketServer, (void *)NULL,
lwipBASIC_SOCKET_SERVER_STACK_SIZE,
lwipBASIC_SOCKET_SERVER_PRIORITY);
#endif

```

File:BasicSOCKET.c

Accion: Agregado

Detalle: Se hace todo el manejo de comunicación por Ethernet pero a nivel de librerías y drivers.

```

#define USART_PROC

#if (SOCKET_USED == 1)

#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "partest.h"
#include "BasicSOCKET.h"

#include "conf_uart_serial.h"
#include "uart_proc.h"
#include "nand_flash_raw.h"
#include "usart.h"
/* Demo includes. */
#include "portmacro.h"

/* lwIP includes. */
#include "lwip/api.h"
#include "lwip/tcpip.h"

```

```

#include "lwip/memp.h"
#include "lwip/stats.h"
#include "lwip/opt.h"
#include "lwip/api.h"
#include "lwip/arch.h"
#include "lwip/sys.h"
#include "lwip/init.h"
#if ( (LWIP_VERSION) == ((1U << 24) | (3U << 16) | (2U << 8) | (LWIP_VERSION_RC)) )
#include "netif/loopif.h"
#endif
#include "lwip/sockets.h"
#include "lwip/smtp.h"
#include <uart.h>

// #define LWIP_COMPAT_SOCKETS 1
#include <lwip/sockets.h>
#include <uart.h>
// #include <lwip/netdb.h>

void my_smtp_result_fn(void *arg, u8_t smtp_result, u16_t srv_err, err_t err);
void my_smtp_result_fn(void *arg, u8_t smtp_result, u16_t srv_err, err_t err)
{
    printf("mail (%p) sent with results: 0x%02x, 0x%04x, 0x%08x\n", arg,
           smtp_result, srv_err, err);
}

#define SENDER_PORT_NUM 6000
#define SENDER_IP_ADDR "192.168.0.102"

#define SERVER_PORT_NUM 6001
#define SERVER_IP_ADDRESS "192.168.0.106"

portTASK_FUNCTION( vBasicSocketServer, pvParameters )
{
    int socket_fd;
    struct sockaddr_in sa,ra;

    int recv_data,sent_data; char data_buffer[80]; /* Creates an TCP socket (SOCK_STREAM) with
    Internet Protocol Family (PF_INET).
    * Protocol family and Address family related. For example PF_INET Protocol Family and AF_INET
    family are coupled.
    */
    for (;;)
    {
        vTaskDelay(2000);
        socket_fd = socket(PF_INET, SOCK_STREAM, 0); //TCP
        // socket_fd = socket(PF_INET, SOCK_DGRAM, 0); //UDP

        if ( socket_fd < 0 )
        {
            printf("socket call failed \n");
            exit(0);
        }

        memset(&sa, 0, sizeof(struct sockaddr_in));
        sa.sin_family = AF_INET;
        sa.sin_len = sizeof(sa);
        // sa.sin_addr.s_addr = inet_addr(SENDER_IP_ADDR);
        sa.sin_addr.s_addr = htonl(INADDR_ANY);

        sa.sin_port = htons(SENDER_PORT_NUM);

        // Bind the TCP socket to the port SENDER_PORT_NUM and to the current
        // machines IP address (Its defined by SENDER_IP_ADDR).
        // Once bind is successful for UDP sockets application can operate
        // on the socket descriptor for sending or receiving data.

        if (bind(socket_fd, (struct sockaddr *)&sa, sizeof(struct sockaddr_in)) == -1)
        {

```

```

        printf("Bind to Port Number %d ,IP address %s
failed\n",SENDER_PORT_NUM,SENDER_IP_ADDR);
        close(socket_fd);
    }
// Receiver connects to server ip-address.
{

    memset(&ra, 0, sizeof(struct sockaddr_in));
    ra.sin_family = AF_INET;
    ra.sin_len = sizeof(ra);
    ra.sin_addr.s_addr = inet_addr(SENDER_IP_ADDRESS);
    ra.sin_port = htons(SENDER_PORT_NUM);
    State_Ethernet=CONNECT_ETH;
    if(connect(socket_fd,(struct sockaddr_in*)&ra,sizeof(struct sockaddr_in)) < 0)
    {
        printf("connect failed \n");
        close(socket_fd);
    }
    else{
        State_Ethernet=LINKED_ETH;
        while(1)
        {

            recv_data = recv(socket_fd,data_buffer,sizeof(data_buffer),0);
            if(recv_data <= 0)
            {
                printf("recv failed \n");
                close(socket_fd);
                break;
            }
            data_buffer[recv_data] = '\0';
            recv_data=0;
            while(data_buffer[recv_data])
            {
                usart_putchar(UART_SERIAL, data_buffer[recv_data]);
                recv_data++;
            }
            //
            printf("received data: %s\n",data_buffer);
            printf("%s",data_buffer);

//-----
//
// State LINKED
//
//-----
                if ((MEMORY.Block_Read!=MEMORY.Block_Write) ||
(MEMORY.Page_Read!=MEMORY.Page_Write)|| (MEMORY.Head_Buffer_Read!=MEMORY.Head_Buffer_Write))
                {

                    nand_flash_raw_read_page(&nf_raw,MEMORY.Block_Write,MEMORY.Page_Write,MEMORY.Buffer_Write,0);

                    MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write]=0x31;
                    sent_data = send(socket_fd,
&MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write],MSGSIZE,0);
                    if(sent_data <= 0 )
                    {
                        printf("send failed\n");
                    }
                    else{

                        MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write]=0x00;

                        nand_flash_raw_write_page(&nf_raw,MEMORY.Block_Write,MEMORY.Page_Write,MEMORY.Buffer_Write,0);

                        MEMORY.Head_Buffer_Write+=PACKETSIZE;
                        if (MEMORY.Head_Buffer_Write >=
nf_raw.model.page_size_in_bytes)
                        {
                            MEMORY.Head_Buffer_Write=0;
                            DEBUG_MICRO(UART_SERIAL, "
CHANGE PAGE!\r\n",15);

                            MEMORY.Page_Write++;
                            if (MEMORY.Page_Write >= 64)
                            {

```



```
#undef USART_PROC
#endif
```

File: init.c

Accion: Agregado

Detalle: Se inicializa la placa con los periféricos y los puertos a utilizar.

```
#include "compiler.h"
#include "board.h"
#include "conf_board.h"
#include "gpio.h"
#include "ioport.h"

void board_init(void)
{
#ifdef CONF_BOARD_KEEP_WATCHDOG_AT_INIT
    /* Disable the watchdog */
    WDT->WDT_MR = WDT_MR_WDDIS;
#endif

    /* GPIO has been deprecated, the old code just keeps it for compatibility.
     * In new designs IOPORT is used instead.
     * Here IOPORT must be initialized for others to use before setting up IO.
     */
    ioport_init();

    /* Configure Power LED */
    // gpio_configure_pin(LED3_GPIO, LED3_FLAGS);
    // gpio_set_pin_high(LED3_GPIO); /* Turned on by default */

    /* Configure User LED pins */
    gpio_configure_pin(LED0_GPIO, LED0_FLAGS);
    gpio_configure_pin(LED1_GPIO, LED1_FLAGS);
    gpio_configure_pin(LED2_GPIO, LED2_FLAGS);

    /* Configure Push Button pins */
    gpio_configure_pin(GPIO_PUSH_BUTTON_1, GPIO_PUSH_BUTTON_1_FLAGS);
    gpio_configure_pin(GPIO_PUSH_BUTTON_2, GPIO_PUSH_BUTTON_2_FLAGS);

#ifdef CONF_BOARD_UART_CONSOLE
    /* Configure UART pins */
    gpio_configure_group(PINS_UART_PIO, PINS_UART, PINS_UART_FLAGS);
#endif

#ifdef CONF_BOARD_USART1_CONSOLE
    /* Configure UART pins */
    gpio_configure_group(PIOA, PIO_PA13A_TXD1, PIO_PERIPH_A | PIO_DEFAULT);
#endif

    /* Configure ADC example pins */
#ifdef CONF_BOARD_ADC
    /* TC TIOA configuration */
    gpio_configure_pin(PIN_TC0_TIOA0, PIN_TC0_TIOA0_FLAGS);

    /* ADC Trigger configuration */
    gpio_configure_pin(PINS_ADC_TRIG, PINS_ADC_TRIG_FLAG);

    /* PWMH0 configuration */
    gpio_configure_pin(PIN_PWMC_PWMH0_TRIG, PIN_PWMC_PWMH0_TRIG_FLAG);
#endif

#ifdef CONF_BOARD_PWM_LED0
    /* Configure PWM LED0 pin */
    gpio_configure_pin(PIN_PWM_LED0_GPIO, PIN_PWM_LED0_FLAGS);
#endif

#ifdef CONF_BOARD_PWM_LED1
    /* Configure PWM LED1 pin */
    gpio_configure_pin(PIN_PWM_LED1_GPIO, PIN_PWM_LED1_FLAGS);
#endif
}
```



```

#ifdef CONF_BOARD_PWM_LED2
    /* Configure PWM LED2 pin */
    gpio_configure_pin(PIN_PWM_LED2_GPIO, PIN_PWM_LED2_FLAGS);
#endif

    /* Configure SPI0 pins */
#ifdef CONF_BOARD_SPI0
    gpio_configure_pin(SPI0_MISO_GPIO, SPI0_MISO_FLAGS);
    gpio_configure_pin(SPI0_MOSI_GPIO, SPI0_MOSI_FLAGS);
    gpio_configure_pin(SPI0_SPCK_GPIO, SPI0_SPCK_FLAGS);

    /**
     * For NPCS 1, 2, and 3, different PINs can be used to access the same
     * NPCS line.
     * Depending on the application requirements, the default PIN may not be
     * available.
     * Hence a different PIN should be selected using the
     * CONF_BOARD_SPI_NPCS_GPIO and
     * CONF_BOARD_SPI_NPCS_FLAGS macros.
     */

#   ifdef CONF_BOARD_SPI0_NPCS0
        gpio_configure_pin(SPI0_NPCS0_GPIO, SPI0_NPCS0_FLAGS);
#   endif

#   ifdef CONF_BOARD_SPI0_NPCS1
#       if defined(CONF_BOARD_SPI0_NPCS1_GPIO) && \
           defined(CONF_BOARD_SPI0_NPCS1_FLAGS)
            gpio_configure_pin(CONF_BOARD_SPI0_NPCS1_GPIO,
                              CONF_BOARD_SPI0_NPCS1_FLAGS);
#       else
            gpio_configure_pin(SPI0_NPCS1_PA29_GPIO,
                              SPI0_NPCS1_PA29_FLAGS);
#       endif
#   endif

#   ifdef CONF_BOARD_SPI0_NPCS2
#       if defined(CONF_BOARD_SPI0_NPCS2_GPIO) && \
           defined(CONF_BOARD_SPI0_NPCS2_FLAGS)
            gpio_configure_pin(CONF_BOARD_SPI0_NPCS2_GPIO,
                              CONF_BOARD_SPI0_NPCS2_FLAGS);
#       else
            gpio_configure_pin(SPI0_NPCS2_PA30_GPIO,
                              SPI0_NPCS2_PA30_FLAGS);
#       endif
#   endif

#   ifdef CONF_BOARD_SPI0_NPCS3
#       if defined(CONF_BOARD_SPI0_NPCS3_GPIO) && \
           defined(CONF_BOARD_SPI0_NPCS3_FLAGS)
            gpio_configure_pin(CONF_BOARD_SPI0_NPCS3_GPIO,
                              CONF_BOARD_SPI0_NPCS3_FLAGS);
#       else
            gpio_configure_pin(SPI0_NPCS3_PA31_GPIO,
                              SPI0_NPCS3_PA31_FLAGS);
#       endif
#   endif
#endif // #ifdef CONF_BOARD_SPI0

    /* Configure SPI1 pins */
#ifdef CONF_BOARD_SPI1
    gpio_configure_pin(SPI1_MISO_GPIO, SPI1_MISO_FLAGS);
    gpio_configure_pin(SPI1_MOSI_GPIO, SPI1_MOSI_FLAGS);
    gpio_configure_pin(SPI1_SPCK_GPIO, SPI1_SPCK_FLAGS);

#   ifdef CONF_BOARD_SPI1_NPCS0
        gpio_configure_pin(SPI1_NPCS0_GPIO, SPI1_NPCS0_FLAGS);
#   endif

#   ifdef CONF_BOARD_SPI1_NPCS1
        gpio_configure_pin(SPI1_NPCS1_GPIO, SPI1_NPCS1_FLAGS);
#   endif

```

```

#  ifdef CONF_BOARD_SPI1_NPCS2
        gpio_configure_pin(SPI1_NPCS2_GPIO, SPI1_NPCS2_FLAGS);
#  endif

#  ifdef CONF_BOARD_SPI1_NPCS3
        gpio_configure_pin(SPI1_NPCS3_GPIO, SPI1_NPCS3_FLAGS);
#  endif
#endif

#ifdef CONF_BOARD_TWI0
    gpio_configure_pin(TWI0_DATA_GPIO, TWI0_DATA_FLAGS);
    gpio_configure_pin(TWI0_CLK_GPIO, TWI0_CLK_FLAGS);
#endif

#ifdef CONF_BOARD_TWI1
    gpio_configure_pin(TWI1_DATA_GPIO, TWI1_DATA_FLAGS);
    gpio_configure_pin(TWI1_CLK_GPIO, TWI1_CLK_FLAGS);
#endif

#ifdef CONF_BOARD_USART_RXD
    /* Configure USART RXD pin */
    gpio_configure_pin(PIN_USART0_RXD_IDX, PIN_USART0_RXD_FLAGS);
#endif

#ifdef CONF_BOARD_USART_TXD
    /* Configure USART TXD pin */
    gpio_configure_pin(PIN_USART0_TXD_IDX, PIN_USART0_TXD_FLAGS);
#endif

#ifdef CONF_BOARD_USART_CTS
    /* Configure USART CTS pin */
    gpio_configure_pin(PIN_USART0_CTS_IDX, PIN_USART0_CTS_FLAGS);
#endif

#ifdef CONF_BOARD_USART_RTS
    /* Configure USART RTS pin */
    gpio_configure_pin(PIN_USART0_RTS_IDX, PIN_USART0_RTS_FLAGS);
#endif

#ifdef CONF_BOARD_USART_SCK
    /* Configure USART synchronous communication SCK pin */
    gpio_configure_pin(PIN_USART0_SCK_IDX, PIN_USART0_SCK_FLAGS);
#endif

#ifdef CONF_BOARD_ADM3312_EN
    /* Configure ADM3312 enable pin */
    gpio_configure_pin(PIN_USART0_EN_IDX, PIN_USART0_EN_FLAGS);
    gpio_set_pin_low(PIN_USART0_EN_IDX);
#endif

#ifdef CONF_BOARD_TFDU4300_SD
    /* Configure IrDA transceiver shutdown pin */
    gpio_configure_pin(PIN_IRDA_SD_IDX, PIN_IRDA_SD_FLAGS);
    gpio_set_pin_low(PIN_IRDA_SD_IDX);
#endif

#ifdef CONF_BOARD_ADM3485_RE
    /* Configure RS485 transceiver ADM3485 RE pin */
    gpio_configure_pin(PIN_RE_IDX, PIN_RE_FLAGS);
    gpio_set_pin_low(PIN_RE_IDX);
#endif

#ifdef CONF_BOARD_SMC_PSRAM
    gpio_configure_pin(PIN_EBI_DATA_BUS_D0, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D1, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D2, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D3, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D4, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D5, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D6, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D7, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D8, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D9, PIN_EBI_DATA_BUS_FLAGS);
    gpio_configure_pin(PIN_EBI_DATA_BUS_D10, PIN_EBI_DATA_BUS_FLAGS);

```

```

gpio_configure_pin(PIN_EBI_DATA_BUS_D11, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D12, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D13, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D14, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D15, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_NRD, PIN_EBI_NRD_FLAGS);
gpio_configure_pin(PIN_EBI_NWE, PIN_EBI_NWE_FLAGS);
gpio_configure_pin(PIN_EBI_NCS0, PIN_EBI_NCS0_FLAGS);
gpio_configure_pin(PIN_EBI_ADDR_BUS_NBS0, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_NBS1, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A1, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A2, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A3, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A4, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A5, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A6, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A7, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A8, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A9, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A10, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A11, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A12, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A13, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A14, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A15, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A16, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A17, PIN_EBI_ADDR_BUS_FLAG1);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A18, PIN_EBI_ADDR_BUS_FLAG2);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A19, PIN_EBI_ADDR_BUS_FLAG2);
gpio_configure_pin(PIN_EBI_ADDR_BUS_A20, PIN_EBI_ADDR_BUS_FLAG2);
#endif // #ifndef CONF_BOARD_SMC_PSRAM

#ifdef CONF_BOARD_HX8347A
/* Configure LCD EBI pins */
gpio_configure_pin(PIN_EBI_DATA_BUS_D0, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D1, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D2, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D3, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D4, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D5, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D6, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D7, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D8, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D9, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D10, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D11, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D12, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D13, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D14, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_DATA_BUS_D15, PIN_EBI_DATA_BUS_FLAGS);
gpio_configure_pin(PIN_EBI_NRD, PIN_EBI_NRD_FLAGS);
gpio_configure_pin(PIN_EBI_NWE, PIN_EBI_NWE_FLAGS);
gpio_configure_pin(PIN_EBI_NCS2, PIN_EBI_NCS2_FLAGS);
gpio_configure_pin(PIN_EBI_LCD_RS, PIN_EBI_LCD_RS_FLAGS);
#endif // #ifndef CONF_BOARD_HX8347A

#ifdef CONF_BOARD_AAT3194
/* Configure Backlight control pin */
gpio_configure_pin(BOARD_AAT31XX_SET_GPIO, BOARD_AAT31XX_SET_FLAGS);
#endif

#ifdef CONF_BOARD_USB_PORT
/* Configure USB_ID (UOTGID) pin */
gpio_configure_pin(USB_ID_GPIO, USB_ID_FLAGS);
/* Configure USB_VBOF (UOTGVBOF) pin */
gpio_configure_pin(USB_VBOF_GPIO, USB_VBOF_FLAGS);
/* Configure FAULT detect pin */
gpio_configure_pin(USB_OVERCURRENT_DETECT_GPIO,
USB_OVERCURRENT_DETECT_FLAGS);
#endif

#ifdef CONF_BOARD_MMA7341L
/* Configure MMA7341L mode set control pin */
gpio_configure_pin(PIN_MMA7341L_MODE, PIN_MMA7341L_MODE_FLAG);

```

```

    /* Configure MMA7341L x,y,z axis output voltage pin */
    gpio_configure_pin(PIN_MMA7341L_X_AXIS, PIN_MMA7341L_X_AXIS_FLAG);
    gpio_configure_pin(PIN_MMA7341L_Y_AXIS, PIN_MMA7341L_Y_AXIS_FLAG);
    gpio_configure_pin(PIN_MMA7341L_Z_AXIS, PIN_MMA7341L_Z_AXIS_FLAG);
#endif

#ifdef CONF_BOARD_ADS7843
    /* Configure Touchscreen SPI pins */
    gpio_configure_pin(BOARD_ADS7843_IRQ_GPIO, BOARD_ADS7843_IRQ_FLAGS);
    gpio_configure_pin(BOARD_ADS7843_BUSY_GPIO, BOARD_ADS7843_BUSY_FLAGS);
    gpio_configure_pin(SPI0_MISO_GPIO, SPI0_MISO_FLAGS);
    gpio_configure_pin(SPI0_MOSI_GPIO, SPI0_MOSI_FLAGS);
    gpio_configure_pin(SPI0_SPCK_GPIO, SPI0_SPCK_FLAGS);
    gpio_configure_pin(SPI0_NPCS0_GPIO, SPI0_NPCS0_FLAGS);
#endif

```

File:opt.h

Accion: Modificado

Detalle: Se habilita el DNS, el reuso de la IP (para multiples conexiones permite reusar la ip de origen) y el debug en determinado puerto.

```

#define LWIP_DNS                1
#define SO_REUSE                1
#define SOCKETS_DEBUG          LWIP_DBG_ON
#define TCP_DEBUG               LWIP_DBG_ON

```

File:conf\_board.h

Accion: Agregado.

Detalle: Se agregan definiciones que se utilizan para habilitar la consola por interfaz UART, la interfaces SPI 0 y 1, para el manejo de la NAND FLASH, y las interfaces UART's.

```

#define CONSOLE_USART1_ID      ID_USART1
#define CONF_BOARD_USART1_CONSOLE

/* Configure SPI0 pins */
#define CONF_BOARD_SPI0
#define CONF_BOARD_SPI0_NPCS0

/* Configure SPI1 pins */
#define CONF_BOARD_SPI1
#define CONF_BOARD_SPI1_NPCS0

/* Configure USART RXD pin */
#define CONF_BOARD_USART_RXD

/* Configure USART TXD pin */
#define CONF_BOARD_USART_TXD

/* Configure USART CTS pin */
// #define CONF_BOARD_USART_CTS

/* Configure USART RTS pin */
#define CONF_BOARD_USART_RTS

/** Support NAND */
#define CONF_BOARD_NAND

```

File:conf\_lwip\_threads.h

Acción: Agregado.

Detalle: Definiciones utilizadas para el manejo de las tareas del sistema operativo.

```

/** define stack size for SOCKET server task */
#define lwipBASIC_SOCKET_SERVER_STACK_SIZE 2048

/** define stack size for SOCKET server task */
#define lwipBASIC_SERIAL_SERVER_STACK_SIZE 256

/** define SOCKET server priority */

```

```

#define lwipBASIC_SOCKET_SERVER_PRIORITY    (tskIDLE_PRIORITY + 4)

/** define SOCKET server priority */
#define lwipBASIC_SERIAL_SERVER_PRIORITY    (tskIDLE_PRIORITY + 5)

```

File:conf\_uart\_serial.h

Acción: Agregado.

Detalle: Estas definiciones sirven para el manejo de las UART's tanto la que se usa para comunicarse con la microestacion como la del modem y ambas se manejan por interrupción.

```

#define USART_SERIAL                USART0
#define USART_SERIAL_ID              ID_USART0
#define USART_SERIAL_ISR_HANDLER    USART0_Handler
#define USART_SERIAL_PIO             PINS_USART_PIO
#define USART_SERIAL_TYPE           PINS_USART_TYPE
#define USART_SERIAL_PINS           PINS_USART_PINS
#define USART_SERIAL_MASK           PINS_USART_MASK
#define USART_SERIAL_BAUDRATE       115200
#define USART_SERIAL_CHAR_LENGTH    US_MR_CHRL_8_BIT
#define USART_SERIAL_PARITY         US_MR_PAR_NO
#define USART_SERIAL_STOP_BIT       US_MR_NBSTOP_1_BIT

#define UART_SERIAL                 UART
#define UART_SERIAL_ID              ID_UART
#define UART_SERIAL_ISR_HANDLER    UART_Handler
#define UART_SERIAL_PIO             PINS_UART_PIO
#define UART_SERIAL_TYPE           PINS_UART_TYPE
#define UART_SERIAL_PINS           PINS_UART_PINS
#define UART_SERIAL_MASK           PINS_UART_MASK
#define UART_SERIAL_BAUDRATE       115200
#define UART_SERIAL_CHAR_LENGTH    US_MR_CHRL_8_BIT
#define UART_SERIAL_PARITY         US_MR_PAR_NO
#define UART_SERIAL_STOP_BIT       US_MR_NBSTOP_1_BIT
#define UART_SERIAL_CHANNEL_MODE    UART_MR_CHMODE_NORMAL
#define UART_SERIAL_MODE           UART_MR_PAR_NO

/** UART Interface */
#define CONF_USART1                 CONSOLE_USART1
/** Baudrate setting */
#define CONF_USART1_BAUDRATE        115200
/** Parity setting */
#define CONF_USART1_PARITY          UART_MR_PAR_NO

```

File:main.c

Accion: Agregado.

Detalle: Se agregan algunas definiciones, se configura la placa, se habilitan interrupciones, se crean las funciones de escritura en la nand flash, la consola, y la tarea de GSM y systick que maneja los timeout de los puertos UART's y los leds.

```

#define TASK_GSM_STACK_SIZE         (128 / sizeof(portSTACK_TYPE))
#define TASK_GSM_STACK_PRIORITY     (tskIDLE_PRIORITY)

/* NAND Flash memory size. */
static uint32_t mem_size;
/* Number of blocks in NAND Flash. */
static uint32_t num_block;
/* Size of one block in the NAND Flash, in bytes. */
static uint32_t block_size;
/* Size of one page in the NAND Flash, in bytes. */
static uint32_t page_size;
/* Number of page per block. */
static uint32_t pages_per_block;

/* NAND Flash device structure. */
//static struct nand_flash_raw nf_raw;
/* Address for transferring command to the NAND Flash. */
static uint32_t cmd_address = BOARD_NF_COMMAND_ADDR;

```

```

/* Address for transferring address to the NAND Flash. */
static uint32_t addr_address = BOARD_NF_ADDRESS_ADDR;
/* Address for transferring data to the NAND Flash. */
static uint32_t data_address = BOARD_NF_DATA_ADDR;

/* Temporary buffer */
uint8_t write_buffer[2048];
uint8_t read_buffer[2048];
uint8_t write_buffer_spare[64];
uint8_t read_buffer_spare[64];

#define PACKETSIZE      128
#define MSGSIZE        69

/**
 * \brief Set peripheral mode for IOPORT pins.
 * It will configure port mode and disable pin mode (but enable peripheral).
 * \param port IOPORT port to configure
 * \param masks IOPORT pin masks to configure
 * \param mode Mode masks to configure for the specified pin (\ref ioport_modes)
 */
#define ioport_set_port_peripheral_mode(port, masks, mode) \
do {\
    ioport_set_port_mode(port, masks, mode);\
    ioport_disable_port(port, masks);\
} while (0)

extern void vApplicationStackOverflowHook(xTaskHandle *pxTask,
signed char *pcTaskName);
extern void vApplicationIdleHook(void);
extern void vApplicationTickHook(void);

extern void xPortSysTickHandler(void);

/** Global timestamp in milliseconds since the start of application */
volatile uint32_t dw_ms_ticks = 0;
volatile uint8_t rx_timeout = 0;

// Hace eco de la comunicaci3n con el concentrador de zigbee si esta bien el paquete sino no
responde nada o coloca TIMEOUT
#ifdef DEBUG_MICROCONTROLADOR
#define DEBUG_MICRO uart_write_buffer
#else
#define DEBUG_MICRO //
#endif

uint32_t nand_flash_raw_write_mem(const struct nand_flash_raw *raw, uint32_t address, uint8_t
*data, uint16_t len)
{
    uint32_t utemp;
    uint32_t block, page;
    volatile uint32_t num_try = 0;

    block=address / (nf_raw.model.block_size_in_kilobytes * 1024);
    page=address % (nf_raw.model.block_size_in_kilobytes * 1024);
    page/=nf_raw.model.page_size_in_bytes;

    smc_nfc_disable_spare_read(SMC);
    smc_nfc_send_command(SMC,
NFCADDR_CMD_NFCCMD      | /* Command. */
NFCADDR_CMD_NFC_READ   | /* NFC read data. */
NFCADDR_CMD_NFCEN     | /* NFC auto R/W is enabled. */
NFCADDR_CMD_CSID(BOARD_NAND_CS) | /* CSID. */
NFCADDR_CMD_ACYCLE_FIVE | /* Number of address cycle. */
NFCADDR_CMD_VCMD2     | /* CMD2 enabled. */
(NAND_COMMAND_READ_2 << 10) | /* CMD2. */
(NAND_COMMAND_READ_1 << 2), /* CMD1. */
address, /* Address cycle 1, 2, 3, 4. */
0 /* Address cycle 0. */
);
/* Wait for the nand to be ready */
num_try = 0;
while (!(smc_nfc_get_status(SMC) & SMC_SR_RB_EDGE0)) &&

```

```

        ((num_try++) < 1000000)) {
    }
    num_try = 0;
    while (((!smc_nfc_get_status(SMC) & SMC_SR_XFRDONE)) &&
        ((num_try++) < 1000000)) {
    }
    /* Read data area if needed */
    //nfc_copy_data_from_internal_sram(data, 0, len);
    //nfc_copy_data_from_internal_sram(data,0,len);
    return 0;
};

/**
 * \brief Handler for Sytem Tick interrupt.
 */
void SysTick_Handler(void)
{
    xPortSysTickHandler();

    dw_ms_ticks++;
    if (dw_ms_ticks>2000)dw_ms_ticks=0;
    // Manejo del Estado INIT

    if (State_Ethernet==INIT_ETH)
    {
        if (dw_ms_ticks<1000)LED_Off(LED0_GPIO);
        else LED_On(LED0_GPIO);
        LED_Off(LED1_GPIO);
    }
    // Manejo del Estado CONNECT
    if (State_Ethernet==CONNECT_ETH)
    {
        if (dw_ms_ticks<1000)LED_Off(LED1_GPIO);
        else LED_On(LED1_GPIO);
        LED_On(LED0_GPIO);
    }
    // Manejo del Estado LINKED
    if (State_Ethernet==LINKED_ETH)
    {
        LED_On(LED0_GPIO);
        LED_On(LED1_GPIO);
    }

    // Manejo del Timeout de recepsion
    if (UART_MICRO.RxState!=WAIT)rx_timeout++;
    else rx_timeout=0;
    if (UART_MICRO.RxState==RX_PROCESSING)
    {
        uint8_t res,val,*ret;

        res=Xor(&UART_MICRO.RX[1],UART_MICRO.RX_Head-6);
        val=strtol(&UART_MICRO.RX[UART_MICRO.RX_Head-5],&ret,16);
        if (val==res)
        {
            DEBUG_MICRO(UART_SERIAL, " MSG OK!\r\n",9);
            // luego eliminarlo y enviar cuando hay diferencia en la memoria
            memcpy(buf_uart_rx_out,UART_MICRO.RX,UART_MICRO.RX_Head);
            buf_uart_rx_out[UART_MICRO.RX_Head+1]=0x00;

            memcpy(&MEMORY.Buffer_Read[MEMORY.Head_Buffer_Read+1],UART_MICRO.RX,MSGSIZE);
            MEMORY.Buffer_Read[MEMORY.Head_Buffer_Read]=0xF0;

            nand_flash_raw_write_page(&nf_raw,MEMORY.Block_Read,MEMORY.Page_Read,MEMORY.Buffer_Read,
0);

            MEMORY.Head_Buffer_Read+=PACKETSIZE;
            if (MEMORY.Head_Buffer_Read >= nf_raw.model.page_size_in_bytes)
            {
                MEMORY.Head_Buffer_Read=0;
                memset(&MEMORY.Buffer_Read,0xFF,sizeof(MEMORY.Buffer_Read));
                DEBUG_MICRO(UART_SERIAL, " CHANGE PAGE READ!\r\n",20);

                nand_flash_raw_read_page(&nf_raw,MEMORY.Block_Read,MEMORY.Page_Read, read_buffer, 0);
                MEMORY.Page_Read++;
            }
        }
    }
}

```

```

        if (MEMORY.Page_Read >= 64)
        {
            MEMORY.Page_Read=0;
            MEMORY.Block_Read++;
            if (MEMORY.Block_Read >= 2048)
            {
                MEMORY.Block_Read=0;
            }
            DEBUG_MICRO(UART_SERIAL, " ERASE BLOCK!\r\n",15);
            nand_flash_raw_erase_block(&nf_raw,MEMORY.Block_Read);
        }
    }

    UART_MICRO.RX_Head=0;
    rx_timeout=0;
    UART_MICRO.RxState=WAIT;

}

if (rx_timeout>RX_TIMEOUT)
{
    UART_MICRO.RX_Head=0;
    rx_timeout=0;
    UART_MICRO.RxState=WAIT;
    DEBUG_MICRO(UART_SERIAL, "TIMEOUT\r\n",9);
}

}

// Tarea del MODEM GSM

static void task_gsm(void *pvParameters)
{
    while(1)
    {
        // Apago y enciendo el modulo GSM por el PIN PE0
        gpio_set_pin_high(POWER_GSM_GPIO);
        delay_s(4);
        gpio_set_pin_low(POWER_GSM_GPIO);
        USART_GSM.Error=ALLOW_ERROR_GSM;
        USART_GSM.State=INIT;
        while(USART_GSM.Error)
        {
            //-----
            //
            // State INIT
            //
            //-----

            while((USART_GSM.State==INIT)&&(USART_GSM.Error))
            {
                // ATE0
                do{usart_write_line(UART_SERIAL,Gsm_Msg_Write[0]);}

            while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,0)==0));
            if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
            // AT&K0
            do{usart_write_line(UART_SERIAL,Gsm_Msg_Write[19]);}

            while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,0)==0));
            if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
            // AT+CGDCONT=1,"IP","gprs.claro.com.ar","",0,0,0,0
            do{usart_write_line(UART_SERIAL,Gsm_Msg_Write[10]);}

            while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,0)==0));
            if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;

            // AT+CREG?
            do{usart_write_line(UART_SERIAL,Gsm_Msg_Write[1]);}

            while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(5,2)==0));
            if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
            // AT+CGACT=1,1
            do{usart_write_line(UART_SERIAL,Gsm_Msg_Write[2]);}

```



```

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(30,0)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    // AT+CGACT?
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[3]);}

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(2,3)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    // AT+CGPADDR=1
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[11]);}

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,0)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    // AT#SGACT=1,1,"clarogprs","clarogprs999
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[12]);}

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(15,4)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    // AT#SCFG=1,1,300,0,600,50
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[20]);}

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,0)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    // AT#SGACT?
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[13]);}

while((USART_GSM.State==INIT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(5,5)==0));
    if ((USART_GSM.State!=INIT)||(!USART_GSM.Error))break;
    if ((USART_GSM.State==INIT)&&(USART_GSM.Error))
        {
            USART_GSM.State=CONNECT;
            USART_GSM.Error=ALLOW_ERROR_GSM;
        }

//-----
//
// State CONNECT
//
//-----
while((USART_GSM.State==CONNECT)&&(USART_GSM.Error))
    {
        // AT#SD=1,0,7000,190.17.248.253,0,0,1
        do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[14]);}

while((USART_GSM.State==CONNECT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(15,0)==0));
    if ((USART_GSM.State!=CONNECT)||(!USART_GSM.Error))break;
    // AT#SS
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[15]);}

while((USART_GSM.State==CONNECT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(5,6)==0));
    if ((USART_GSM.State!=CONNECT)||(!USART_GSM.Error))break;
    // AT#SEND=1
    do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[16]);}

while((USART_GSM.State==CONNECT)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,7)==0));
    if ((USART_GSM.State!=CONNECT)||(!USART_GSM.Error))break;
    usart_write_line(USART_SERIAL,Gsm_Msg_Write[18]);
    usart_write_line(USART_SERIAL,Gsm_Msg_Write[17]);
    if ((USART_GSM.State==CONNECT)&&(USART_GSM.Error))
        {
            USART_GSM.State=LINKED;
            USART_GSM.Error=ALLOW_ERROR_GSM;
        }

}

//-----
//
// State LINKED
//
//-----
while((USART_GSM.State==LINKED)&&(USART_GSM.Error))
    {
//
        while(UART_MICRO.TxState==EMPTY);
    }

```

```

        while((MEMORY.Block_Read==MEMORY.Block_Write)&&(MEMORY.Page_Read==MEMORY.Page_Write)&&(M
MEMORY.Head_Buffer_Read==MEMORY.Head_Buffer_Write));
            // AT#SEND
            do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[16]);}

        while((USART_GSM.State==LINKED)&&(USART_GSM.Error)&&(Gsm_Proc_Rec(3,7)==0));
            if ((USART_GSM.State!=LINKED)||(!USART_GSM.Error))break;
            if ((USART_GSM.State==LINKED)&&(USART_GSM.Error))
                {

        nand_flash_raw_read_page(&nf_raw,MEMORY.Block_Write,MEMORY.Page_Write,MEMORY.Buffer_Writ
e,0);
            MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write]=0x31;
            uart_write_buffer(USART_SERIAL,
&MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write],MSGSIZE+1);
            usart_write_line(USART_SERIAL,Gsm_Msg_Write[17]);
            if (Gsm_Proc_Rec(5,0)==1)
                {

        MEMORY.Buffer_Write[MEMORY.Head_Buffer_Write]=0x00;

        nand_flash_raw_write_page(&nf_raw,MEMORY.Block_Write,MEMORY.Page_Write,MEMORY.Buffer_Wri
te,0);
            MEMORY.Head_Buffer_Write+=PACKETSIZE;
            if (MEMORY.Head_Buffer_Write >=
nf_raw.model.page_size_in_bytes)
                {
                    MEMORY.Head_Buffer_Write=0;
                    DEBUG_MICRO(UART_SERIAL, "
CHANGE PAGE!\r\n",15);
                    MEMORY.Page_Write++;
                    if (MEMORY.Page_Write >= 64)
                        {
                            MEMORY.Page_Write=0;
                            MEMORY.Block_Write++;
                            if (MEMORY.Block_Write
                                >= 2048)
                                {
                                    MEMORY.Block_Write=0;
                                }
                            }
                    }
                }
            }
        }
    }
}

/**
 * \brief Configure the console UART.
 */
static void configure_console(void)
{
    const usart_serial_options_t usart_serial_options = {
        .baudrate = CONF_USART1_BAUDRATE,
        .paritytype = CONF_USART1_PARITY
    };

    /* Configure UART pins */
    ioport_set_port_peripheral_mode(PINS_UART0_PORT, PIO_PA13A_TXD1,PIO_PA13A_TXD1);

    /* Configure console UART. */
    sysclk_enable_peripheral_clock(CONSOLE_USART1_ID);
    stdio_serial_init(CONF_USART1, &usart_serial_options);

    const sam_usart_opt_t usart_console_settings = {
        USART_SERIAL_BAUDRATE,
        USART_SERIAL_CHAR_LENGTH,
        USART_SERIAL_PARITY,

```

```

        USART_SERIAL_STOP_BIT,
        US_MR_CHMODE_NORMAL
};

usart_init_rs232(USART1, &usart_console_settings, sysclk_get_main_hz());
usart_enable_tx(USART1);

}

/**
 * \brief getting-started Application entry point.
 *
 * \return Unused (ANSI-C compatibility).
 */
int main(void)
{
    uint16_t block, page;
    uint32_t i;

    uint32_t error = 0;

    board_init();

    /* Disable the watchdog */
    WDT->WDT_MR = WDT_MR_WDDIS;

    /* Initilize the SAM system */
    sysclk_init();

    memset(&MEMORY.Buffer_Read,0xFF,sizeof(MEMORY.Buffer_Read));
    memset(&MEMORY.Buffer_Write,0xFF,sizeof(MEMORY.Buffer_Write));

    /* Output example information */
    memset(&nf_raw, 0, sizeof(nf_raw));

    if (nand_flash_raw_initialize(&nf_raw, 0, cmd_address,addr_address, data_address))
    {
        printf("-E- Device Unknown\n\r");
        return 0;
    }

    // printf("-I- NAND Flash driver initialized\n\r");

    /* Get device parameters */
    mem_size = nand_flash_model_get_device_size_in_bytes(&nf_raw.model);
    block_size = nand_flash_model_get_block_size_in_bytes(&nf_raw.model);
    num_block = nand_flash_model_get_device_size_in_blocks(&nf_raw.model);
    page_size = nand_flash_model_get_page_data_size(&nf_raw.model);
    pages_per_block = nand_flash_model_get_block_size_in_pages(&nf_raw.model);

    // Erase all block and use the last good block for read/write test
    for (i =0; i < num_block; i++) {
        error = nand_flash_raw_erase_block(&nf_raw, i);

        if (error == NAND_COMMON_ERROR_BADBLOCK) {
            printf("-E- Block %u is BAD block. \n\r", i);
            nand_flash_raw_read_page(&nf_raw, i, 0, read_buffer, 0);
        } else {
            block = i;
        }
    }

    // Initialize the console uart
    configure_console();

    /* Configure the pins connected to LEDs as output and set their
     * default initial state to high (LEDs off).
     */
    ioport_set_pin_dir(LED0_GPIO, IOPORT_DIR_OUTPUT);
    ioport_set_pin_level(LED0_GPIO, LED0_INACTIVE_LEVEL);
    ioport_set_pin_dir(LED1_GPIO, IOPORT_DIR_OUTPUT);
    ioport_set_pin_level(LED1_GPIO, LED1_INACTIVE_LEVEL);
    ioport_set_pin_dir(LED2_GPIO, IOPORT_DIR_OUTPUT);
    ioport_set_pin_level(LED2_GPIO, LED2_INACTIVE_LEVEL);

```

```

// Configure Power GSM
gpio_configure_pin(POWER_GSM_GPIO, POWER_GSM_FLAGS);
gpio_set_pin_high(POWER_GSM_GPIO); /* Turned on by default */

/* Configure USART RXD pin */
gpio_configure_pin(PIN_USART0_RXD_IDX, PIN_USART0_RXD_FLAGS);
/* Configure USART TXD pin */
gpio_configure_pin(PIN_USART0_TXD_IDX, PIN_USART0_TXD_FLAGS);
/* Configure ADM3312 enable pin */
gpio_configure_pin(PIN_USART0_EN_IDX, PIN_USART0_EN_FLAGS);
gpio_set_pin_low(PIN_USART0_EN_IDX);
/* Configure RTS enable pin */

gpio_configure_pin(PIN_USART0_RTS_IDX, PIN_USART0_RTS_FLAGS);
gpio_set_pin_low(PIN_USART0_RTS_IDX);
gpio_set_pin_high(PIN_USART0_RTS_IDX);
gpio_set_pin_low(PIN_USART0_RTS_IDX);
gpio_set_pin_high(PIN_USART0_RTS_IDX);
gpio_set_pin_low(PIN_USART0_RTS_IDX);

const sam_usart_opt_t usart_gsm_settings = {
    USART_SERIAL_BAUDRATE,
    USART_SERIAL_CHAR_LENGTH,
    USART_SERIAL_PARITY,
    USART_SERIAL_STOP_BIT,
    US_MR_CHMODE_NORMAL };

sysclk_enable_peripheral_clock(USART_SERIAL_ID);

usart_init_rs232(USART_SERIAL, &usart_gsm_settings, sysclk_get_main_hz());
usart_enable_tx(USART_SERIAL);
usart_enable_rx(USART_SERIAL);

usart_enable_interrupt(USART_SERIAL, US_IER_RXRDY);
NVIC_EnableIRQ(USART0_IRQn);

/* Configure UART pins */
ioport_set_port_peripheral_mode(PINS_UART0_PORT, PINS_UART0,
PINS_UART0_MASK);

/* Configure UART. */
const sam_uart_opt_t uart_micro_settings =
{ sysclk_get_cpu_hz(), USART_SERIAL_BAUDRATE, USART_SERIAL_MODE | USART_MR_CHMODE_NORMAL };

sysclk_enable_peripheral_clock(UART_SERIAL_ID);

uart_init(UART_SERIAL, &uart_micro_settings);
uart_enable_tx(UART_SERIAL);
uart_enable_rx(UART_SERIAL);

uart_enable_interrupt(UART_SERIAL, UART_IER_RXRDY);
NVIC_EnableIRQ(UART_IRQn);

printf("-- Gsm Project --\n\r");
printf("-- %s\n\r", BOARD_NAME);
printf("-- Compiled: %s %s --\n\r", __DATE__, __TIME__);
delay_s(1);

// Ethernet Project

State_Ethernet=INIT_ETH;

// usart_write_line(USART1,"Comienzo de comunicacion\n");

/* Output demo infomation. */
printf("-- FreeRTOS with lwIP Example --\n\r");
printf("-- %s\n\r", BOARD_NAME);
printf("-- Compiled: %s %s --\n\r", __DATE__, __TIME__);

/* Create task to make led blink */
if (xTaskCreate(task_gsm, "Gsm", TASK_GSM_STACK_SIZE, NULL,
TASK_GSM_STACK_PRIORITY, NULL) != pdPASS) {
    printf("Failed to create test GSM task\r\n");
}

```

```

}
/* Start the ethernet tasks */
vStartEthernetTaskLauncher( configMAX_PRIORITIES );

/* Start FreeRTOS */
vTaskStartScheduler();

/* Will only reach here if there was insufficient memory to create the idle task */
return 0;
// Insert application code here, after the board has been initialized.
}

```

File:uart\_proc.c

Accion: Agregado.

Detalle: Se realiza el manejo de las UART's y las definiciones de los comandos al modem gsm, las tareas de interrupciones de los puertos UART y algunas funciones que sirven como la comprobación de los mensajes y su chequeo (xor, HexToBcd, strstr\_T).

```

// Hace eco de la comunicacion con el concentrador de zigbee si esta bien el paquete sino no responde nada o coloca TIMEOUT

```

```

#define DEBUG_MICRO uart_write_buffer
// Public variable

```

```

uint8_t *Gsm_Msg_Write[]={
/* 0 */ "ATE0\r\n",
/* 1 */ "AT+CREG?\r\n",
/* 2 */ "AT+CGACT=1,1\r\n",
/* 3 */ "AT+CGACT?\r\n",
/* 4 */ "AT+MIPCALL=1,\"internet.gprs.unifon.com.ar\", \"wap\", \"wap\"\r\n",
/* 5 */ "AT+MIPODM=1,6001,\"190.16.196.191\",6001,0\r\n",
/* 6 */ "AT+MIPCLOSE=1\r\n",
/* 7 */ "+++",
/* 8 */ "ATO\r\n",
/* 9 */ "AT+MIPCALL?\r\n",
#ifdef CLARO
/* 10 */ "AT+CGDCONT=1,\"IP\", \"gprs.claro.com.ar\", \"0.0.0.0\",0,0\r\n",
#endif
#ifdef MOVISTAR
/* 10 */ "AT+CGDCONT=1,\"IP\", \"wap.gprs.unifon.com.ar\", \"0.0.0.0\",0,0\r\n",
#endif
#ifdef PERSONAL
/* 10 */ "AT+CGDCONT=1,\"IP\", \"gprs.personal.com\", \"0.0.0.0\",0,0\r\n",
#endif
/* 11 */ "AT+CGPADDR=1\r\n",
#ifdef CLARO
/* 12 */ "AT#SGACT=1,1,\"clarogprs\", \"clarogprs999\r\n",
#endif
#ifdef MOVISTAR
/* 12 */ "AT#SGACT=1,1,\"wap\", \"wap\r\n",
#endif
#ifdef PERSONAL
/* 12 */ "AT#SGACT=1,1,\"gprs\", \"gprs\r\n",
#endif
/* 13 */ "AT#SGACT?\r\n",
/* 14 */ "AT#SD=1,0,7000,190.16.196.191,0,0,1\r\n",
/* 15 */ "AT#SS\r\n",
/* 16 */ "AT#SEND=1\r\n",
/* 17 */ "\x1A",
/* 18 */ "We are Alive\r\n",
/* 19 */ "AT&K0\r\n",
/* 20 */ "AT#SCFG=1,1,300,0,600,50\r\n"

```

```

};

```

```

uint8_t *Gsm_Msg_Read[]={
/* 0 */ "OK\r\n",
/* 1 */ "ERROR\r\n",
/* 2 */ "+CREG: 0,1\r\n",
/* 3 */ "+CGACT: 1,1\r\n",

```

```

        /* 4 */"#SGACT:",
        /* 5 */"#SGACT: 1,1\r\n",
        /* 6 */"#SS: 1,2",
        /* 7 */">",
};

// Public Function

void uart_write_line(Uart *p_uart, const char *string)
{
    while (*string != '\0') {
        while(!(p_uart->UART_SR & UART_SR_TXRDY));
        p_uart->UART_THR = *string;
        string++;
    }
}

uint32_t uart_write_buffer(Uart *p_uart, uint8_t *buffer, uint16_t n)
{
    /* Check if the transmitter is ready */
    while(n)
    {
        while(!(p_uart->UART_SR & UART_SR_TXRDY));
        n--;
        p_uart->UART_THR = *buffer;
        buffer++;
    }
    /* Send character */
    return 0;
}

void UART_SERIAL_ISR_HANDLER(void)
{
    uint32_t dw_status = uart_get_status(UART_SERIAL);

    uart_read(UART_SERIAL, &UART_MICRO.RX[UART_MICRO.RX_Head]);

    if (dw_status & UART_SR_RXRDY)
    {
        uint32_t received_byte;
        if (UART_MICRO.RxState==WAIT_CMD)
        {
            if
            ((UART_MICRO.RX[UART_MICRO.RX_Head]==CMD_D) || (UART_MICRO.RX[UART_MICRO.RX_Head]==CMD_C) || (UART_M
            ICRO.RX[UART_MICRO.RX_Head]==CMD_F))
            {
                UART_MICRO.RxState=RECEIVING;
            }
            else{
                UART_MICRO.RxState=WAIT;
                UART_MICRO.RX_Head=0;
                UART_MICRO.RX[0]=0;
            }
        }
        if
        ((UART_MICRO.RX[UART_MICRO.RX_Head]==START_STREAM)&&(UART_MICRO.RxState==WAIT))
        {
            UART_MICRO.RX_Head++;
            UART_MICRO.RxState=WAIT_CMD;
        }
        if (UART_MICRO.RxState==WAIT_W)
        {
            if (UART_MICRO.RX[UART_MICRO.RX_Head]==POST_END)
            {
                UART_MICRO.RX_Head++;
                memcpy(UART_MICRO.TX, UART_MICRO.RX, UART_MICRO.RX_Head);
                UART_MICRO.TX_Head=UART_MICRO.RX_Head;
                UART_MICRO.TxState=PROCESS;
                UART_MICRO.RX[UART_MICRO.RX_Head]=0x00;
                DEBUG_MICRO(UART_SERIAL, &UART_MICRO.TX, UART_MICRO.TX_Head);
            }
        }
    }
}

```

```

        UART_MICRO.RxState=RX_PROCESSING;
    }
    else{
        UART_MICRO.RxState=RECEIVING;
    }
}

    if
((UART_MICRO.RX[UART_MICRO.RX_Head]==END_STREAM)&&(UART_MICRO.RxState==RECEIVING))
    {
        UART_MICRO.RxState=WAIT_W;
        UART_MICRO.RX_Head++;
    }
    if (UART_MICRO.RxState==RECEIVING)UART_MICRO.RX_Head++;
    if (UART_MICRO.RX_Head>=BUFFER_RX_UART){
UART_MICRO.RX_Head=0;UART_MICRO.RxState=WAIT;}
}
if (dw_status & UART_SR_OVRE)
{uart_reset_status(UART_SERIAL);UART_MICRO.RX_Head=0;UART_MICRO.RxState=WAIT;}
if (dw_status & UART_SR_FRAME)
{uart_reset_status(UART_SERIAL);UART_MICRO.RX_Head=0;UART_MICRO.RxState=WAIT;}
if (dw_status & UART_SR_PARE)
{uart_reset_status(UART_SERIAL);UART_MICRO.RX_Head=0;UART_MICRO.RxState=WAIT;}
}

void USART_SERIAL_ISR_HANDLER(void)
{
    uint32_t dw_status = usart_get_status(USART_SERIAL);

    if (dw_status & US_CSR_RXRDY)
    {
        uint32_t received_byte;
        usart_read(USART_SERIAL, &USART_GSM.RX[USART_GSM.RX_Head]);
        USART_GSM.RX_Head++;
        if (USART_GSM.RX_Head>=BUFFER_RX_GSM) USART_GSM.RX_Head=0;
    }
}

uint8_t* strstr_T(uint8_t *S1, uint8_t *S2,uint16_t LenS1,uint16_t LenS2)
{
    uint8_t *Puint8_t=0,*SS2,*SS1;
    uint16_t Equal=0;

    SS2=S2;
    SS1=S1;

    while(LenS1)
    {
        LenS1--;
        if (*SS2==*SS1)
        {
            if (Equal);
            else Puint8_t=SS1;
            Equal++;SS2++;
            if (Equal>=LenS2)
                return Puint8_t;
        }
        else {
            if (Equal){SS1--;LenS1++;}
            Equal=0;
            SS2=S2;
        }
        SS1++;
    }
    return 0;
}

uint8_t Gsm_Proc_Rec(uint8_t delay_seg,uint8_t num)
{
    uint8_t *Pun;
    uint16_t loop;

```

```

USART_GSM.RX_DelayIn=delay_seg;/**TICK_DIV;
//USART_GSM.RX_Head=0;
while(USART_GSM.RX_DelayIn)
{
    delay_s(1);
    USART_GSM.RX_DelayIn--;
    loop=num;
//    for (loop=0;loop<(sizeof(Gsm_Msg_Read)>>2);loop++)
    {
        Pun=Gsm_Msg_Read[loop];
        Pun=strstr_T(USART_GSM.RX,(uint8_t
*)Gsm_Msg_Read[loop],USART_GSM.RX_Head,strlen(Gsm_Msg_Read[loop]));
//        Pun=strstr(USART_GSM.RX,"OK\r\n");
        if ((Pun)&&(USART_GSM.RX_Head))
            {
                USART_GSM.Error=ALLOW_ERROR_GSM;
                USART_GSM.RX_Head=0;
//                printf("%s",Gsm_Msg_Read[loop]);
                *Pun=0x00;
                return 1;
//                break;
            }
    }
//while(USART_GSM.RX_TimeOut);
if (USART_GSM.Error)USART_GSM.Error--;
if (!USART_GSM.Error)USART_GSM.State=RESET;
return 0;
}

uint8_t Xor(uint8_t *Data, uint16_t Len)
{
    uint8_t result=0;

    while(Len)
        {
            result^=*Data;
            Data++;
            Len--;
        }
    return result;
}

uint8_t HexToBcd(uint8_t hex)
{
    uint8_t decena;
    uint8_t unidad;

    unidad=hex%10;
    decena=hex/10;
    unidad+=decena<<4;
    return unidad;
}

```

#### 4.5- Pruebas y ensayos.

Para realizar las pruebas se utilizó el software Hercules, el cual se puede descargar en forma gratuita de [http://www.hw-group.com/products/hercules/index\\_es.html](http://www.hw-group.com/products/hercules/index_es.html). El mismo permite crear un cliente o servidor tanto de TCP como de UDP, como también usarlo para el manejo del puerto serie. Para las pruebas realizadas tanto el kit de desarrollo, cliente, como la PC donde estaba corriendo el software Hercules, servidor, se encontraban en la misma red, por lo tanto las direcciones IP de ellos son similares. Es importante destacar que las pruebas podrían haber sido realizadas instalando el kit de desarrollo y la PC en redes separadas y comunicarse y los resultados hubieran sido los mismos.



Las direcciones IP deben ser configuradas en el archivo BasicSocket.c con las siguientes direcciones:

- IP Kit: 192.168.0.105 Port 6000
- IP Server: 192.168.0.106 Port 6001

Usualmente los Router están configurados para asignar las direcciones IP a los clientes en forma dinámica a través del protocolo DHCP, asignando a un mismo cliente la misma dirección IP cada vez que se conecta, por lo tanto hasta que no se conecta por primera vez el kit a la red no se sabra con que dirección IP va trabajar, y por lo tanto que dirección IP hay que configurar en el kit. A continuación se puede ver la información enviada por el kit a través de la interfaz de debug una vez que se conecta a Ethernet.

```
10/01/2016 16:01:34.368 [RX] - -- Gsm Project --  
-- SAM3X-EK  
-- Compiled: Jan 10 2016 15:27:26 --  
  
10/01/2016 16:01:35.374 [RX] - -- FreeRTOS with lwIP Example --  
-- SAM3X-EK  
-- Compiled: Jan 10 2016 15:27:26 --  
  
10/01/2016 16:01:39.968 [RX] - LwIP: DHCP Started  
  
10/01/2016 16:01:40.976 [RX] - Network up IP=192.168.0.105
```

Obtenida la IP se configuro el kit la IP 192.168.0.105 puerto socket 6000 y el servidor con la IP 192.168.0.106 puerto socket 6001.

A continuación se ve la información que el kit envía a través de la interfaz de debug una vez que comienza a correr la tarea de SOCKET.

```
10/01/2016 16:01:41.989 [RX] - lwip_socket(PF_INET, SOCK_STREAM, 0) =  
0  
lwip_bind(0, addr=0.0.0.0 port=6000)  
lwip_bind(0) succeeded  
lwip_connect(0, addr=192.168.0.106 port=6001)  
  
10/01/2016 16:01:54.246 [RX] - lwip_connect(0) failed, err=-10  
connect failed  
lwip_close(0)  
  
10/01/2016 16:01:56.230 [RX] - lwip_socket(PF_INET, SOCK_STREAM, 0) =  
0  
lwip_bind(0, addr=0.0.0.0 port=6000)  
lwip_bind(0) succeeded  
lwip_connect(0, addr=192.168.0.106 port=6001)
```

Se intenta conectar con la IP del servidor y logra realizarlo en el segundo intento. Una vez realizada la conexión la aplicación queda esperando recibir un dato a través de la interfaz serial, proveniente de la microestación, o recibir un mensaje a través de la interfaz de Ethernet, proveniente del servidor.

Recibido un mensaje de la microestacion, la rutina de servicio de interrupción lo analizara y almacenara para su posterior envio al servidor. A continuación se ve la recepción de un mensaje por la interfaz serie.

```
16:02:14.066 [TX] -  
$D,ESTAC_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,33,#W
```

Si mensaje esta correcto y se devuelve el mensaje a la microestacion mas el agregado de MSG OK!.

```
16:02:14.094 [RX] -  
$D,ESTAC_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,33,#W  
MSG OK!
```

Recibido un mensaje desde el servidor, se envía los datos pendientes en memoria.

```
16:02:38.281 [RX] - C160  
10/01/2016 16:02:38.305 [RX] - lwip_recvfrom: netconn_recv err=0,  
netbuf=0x20075f38  
lwip_recvfrom: buflen=4 len=80 off=0 sock->lastoffset=0  
lwip_recvfrom(0): addr=192.168.0.106 port=6001 len=4  
lwip_recvfrom: deleting netbuf=0x20075f38
```

C160

```
lwip_send(0, data=0x200771c4, size=68, flags=0x0)  
lwip_send(0) err=0 size=68  
lwip_recvfrom(0, 0x200829f8, 80, 0x0, ..)  
lwip_recvfrom: top while sock->lastdata=0x0
```

La figura N°4.4 se ve el mensaje enviado por la microestacion y el comando C160 de pedido de mensajes por parte del servidor.

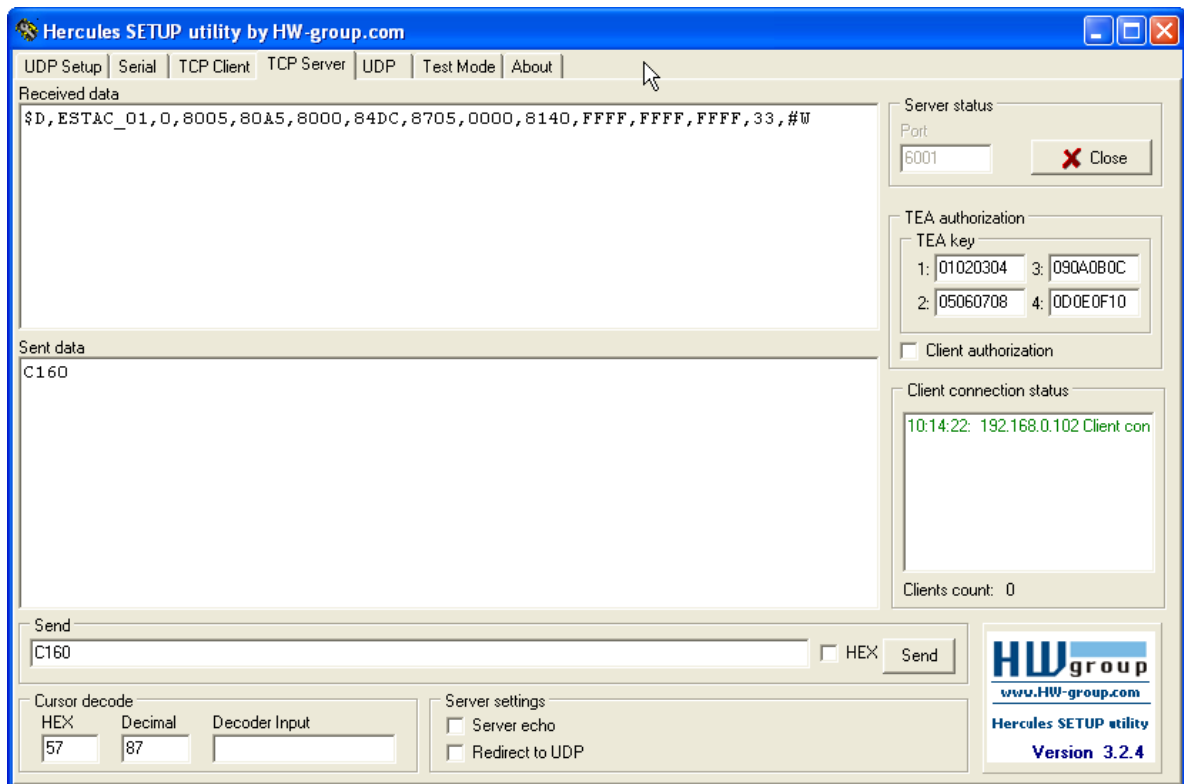


Figura N°4.4. Terminal de Debug - Comunicación Ethernet.

En el caso que la conexión al servidor a través de Ethernet no se pueda realizar, se dispara la tarea que intentara realizar la conexión a través de GSM/GPRS. A continuación se muestra se el log de la secuencia de comandos enviados al modulo y sus respuestas.

Se comienza configurando el modulo GSM/GPRS, se elimina el eco de caracteres, se deshabilita el control de flujo de la interfaz serie, se configura el contexto para realizar

la conexión GPRS con la prestadora de servicio, se verifica que este registrado en la red GSM.

```
14:45:24.367 [RX] - ATE0  
ATE0
```

OK

```
14:45:25.374 [RX] - AT&K0
```

OK

```
14:45:26.397 [RX] - AT+CGDCONT=1,IP,gprs.claro.com.ar,0.0.0.0,0,0
```

OK

```
14:45:27.436 [RX] - AT+CREG?
```

```
+CREG: 0,2
```

OK

```
14:45:32.504 [RX] - AT+CREG?
```

```
+CREG: 0,1
```

OK

La secuencia de comandos que se muestran a continuación están asociados a la obtención de una IP dentro de la red privada de la prestadora de servicio.

```
14:45:33.527 [RX] - AT+CGACT=1,1
```

```
14:45:34.199 [RX] -
```

OK

```
AT+CGACT?
```

```
+CGACT: 1,1
```

OK

```
14:45:35.541 [RX] - AT+CGPADDR=1
```

```
+CGPADDR: 1,"186.123.77.186"
```

OK

```
14:45:36.565 [RX] - AT#SGACT=1,1,clarogprs,clarogprs999
```

```
#SGACT: 186.123.77.186
```

OK

```
14:45:37.588 [RX] - AT#SCFG=1,1,300,0,600,50
```

OK

```
14:45:38.595 [RX] - AT#SGACT?
```

```
#SGACT: 1,1
```

OK

Obtenida la IP se intenta abrir un puerto en el servidor.

```
14:45:39.618 [RX] - AT#SD=1,0,7000,181.46.152.110,0,0,1
```

```
14:45:40.290 [RX] -
OK
AT#SS

#SS: 1,2,186.123.77.186,7640,181.46.152.110,7000
#SS: 2,0
#SS: 3,0
#SS: 4,0
#SS: 5,0
#SS: 6,0

OK
```

Una vez establecida la conexión se envía un mensaje de bienvenida.

```
14:45:41.638 [RX] - AT#SSEND=1

>
14:45:42.662 [RX] - We are Alive
_AT#SSEND=1

OK
```

Mientras no tenga datos para enviar, el programa quedara en un bucle chequeando la conexión y verificando si tiene datos para enviar, tal como se puede ver en el diagrama de flujo de la Figura 4.3 antes presentado.

Como en nuestro caso se enviaron 4 paquetes mientras estaba comenzado la secuencia de inicialización del modulo, estos se almacenaron.

```
14:45:25.889 [TX] -
$D,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,31,#W
14:45:25.916 [RX] -
$D,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,31,#W
MSG OK!

14:45:26.512 [TX] -
$D,ESTAC_02,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,30,#W
14:45:26.524 [RX] -
$D,ESTAC_02,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,30,#W
MSG OK!

14:45:27.161 [TX] -
$D,ESTAC_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,33,#W
14:45:27.189 [RX] -
$D,ESTAC_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,33,#W
MSG OK!

14:45:27.947 [TX] -
$C,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,36,#W
14:45:27.956 [RX] -
$C,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,36,#W
MSG OK!
```

Y al llegar al punto antes mencionado se realizó la transmisión.

```
14:45:45.683 [RX] - AT#SSEND=1

>
14:45:46.690 [RX] -
$D,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,31,#W_
OK

14:45:47.719 [RX] - AT#SSEND=1

>
```

```

14:45:48.736 [RX] -
$D,ESTAC_02,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,30,#W_
OK

14:45:49.742 [RX] - AT#SSEND=1

>
14:45:50.760 [RX] -
$D,ESTAC_01,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,33,#W_
OK

14:45:51.767 [RX] - AT#SSEND=1

>
14:45:52.785 [RX] -
$C,ESTAC_03,0,8005,80A5,8000,84DC,8705,0000,8140,FFFF,FFFF,FFFF,36,#W_
OK

```

La figura N°4.5 muestra los mensajes enviados por la estación colectora al servidor.

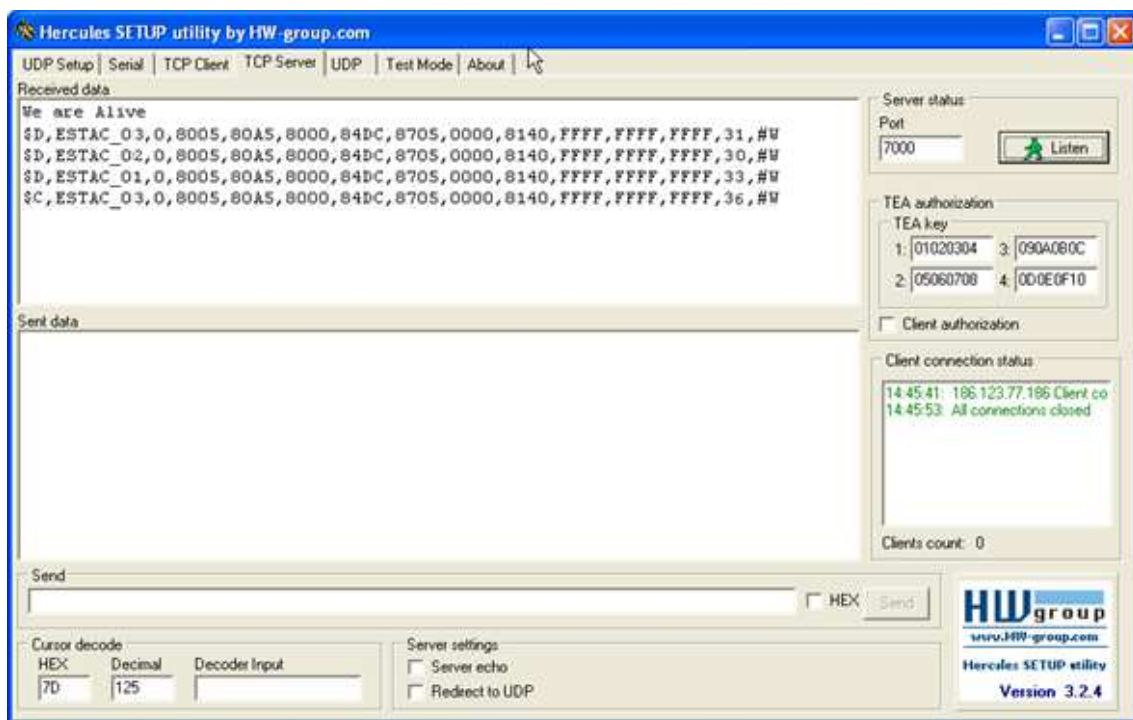


Figura N°4.5. Terminal de Debug – Comunicación GSM/GPRS.

#### 4.6 –Conclusiones

Se lograron la mayoría de los objetivos planteados, obteniendo un sistema con un alto grado de flexibilidad.

Se migro el código desarrollado previamente para el modulo Motorola G24 a un modulo Telit GL865, código el cual es compatible con la versión de tecnología 3G del modulo (UL865). Se implemento la generación de buffer de almacenamiento, la configuración de datos de conexión y el mejoramiento de las rutinas asociadas a la obtención de IP por DHCP, con la salvedad del reconocimiento automático de la IP. Se integro en un solo programa el envío de datos por medio de Ethernet o por GSM/GPRS.

A raíz que la migración del código de un modulo celular a otro llevo más tiempo que el esperado, ya que la mayoría de los comandos que se deben implementar para el envío de datos por GPRS no son compatible, no se profundizo en el estudio del stack SMTP para el envío de datos con seguridad SSL a través de Ethernet, sin embargo se estudiaron los comandos AT para realizar dicho envío a través de la comunicación GSN/GPRS, quedando pendiente su implementación.

De forma similar no se llevo a profundizar en el estudio del stack TCP de forma que no sea necesario el envío de un paquete desde el server para recibir los datos del cliente, por lo tanto se implementa que si la comunicación sea a través de Ethernet se conecte a un puerto en el servidor, el cual estará enviando una consulta periódicamente. Esto se puede apreciar en los ensayos realizados.

## 5- Software

A continuación describe la arquitectura del software involucrado en el proyecto: el mismo constará de dos partes independientes llamadas Servidor de Datos (que adquiriría información 24hs al día) y Cliente de Operaciones disponible al usuario final.

Este último permitirá el tratamiento y adecuación de la información a los distintos requerimientos del usuario, brindando flexibilidad para ser aplicado a diversas aplicaciones agroindustriales.

### 5.1- Tipos de arquitecturas de comunicación

A continuación se describirán los tres tipos de arquitecturas que se pretenden implementar:

- Comunicación local por puerto serie RS232
- Comunicación remota utilizando un servidor de correo electrónico
- Comunicación remota por TCP/IP

#### 5.1.1- Esquema de la Arquitectura por Puerto Serie RS232

Bosquejo de lo que se quiere implementar (Figura N°5.1):

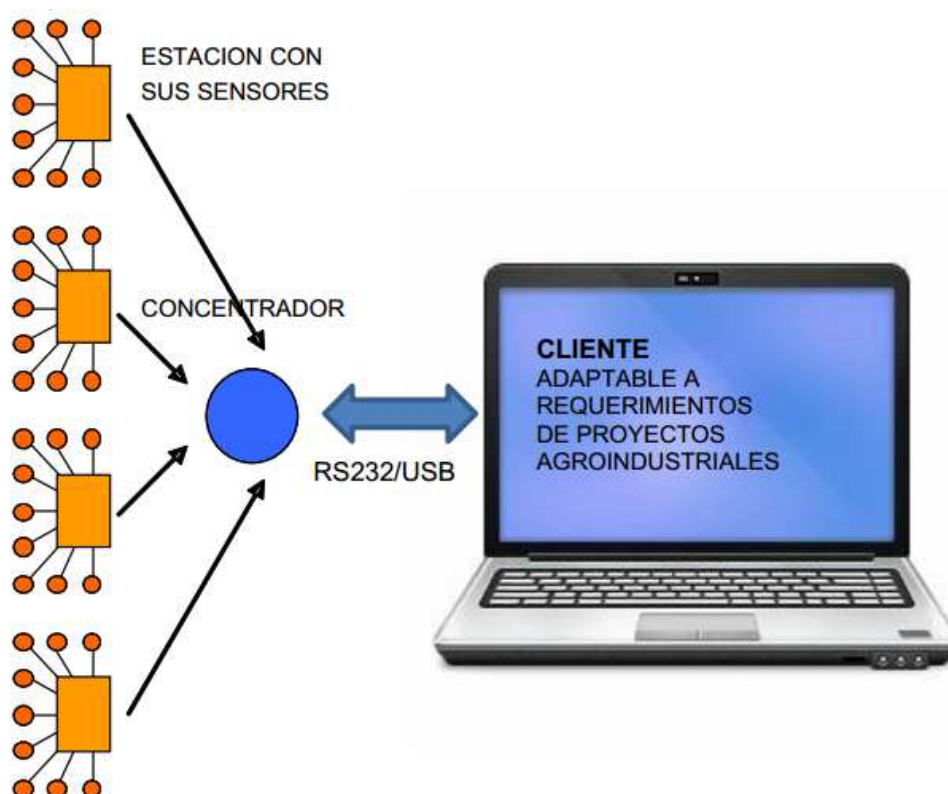


Figura N°5.1- Esquema de arquitectura del puerto serie RS232

Comunicación bidireccional entre aplicación "Cliente" y la estación Concentradora.

El protocolo se detalla en un apartado más adelante en este mismo texto.

### 5.1.2- Esquema de la Comunicación mediante un Servidor de Correo

Bosquejo de lo que se quiere implementar (Figura N°5.2):

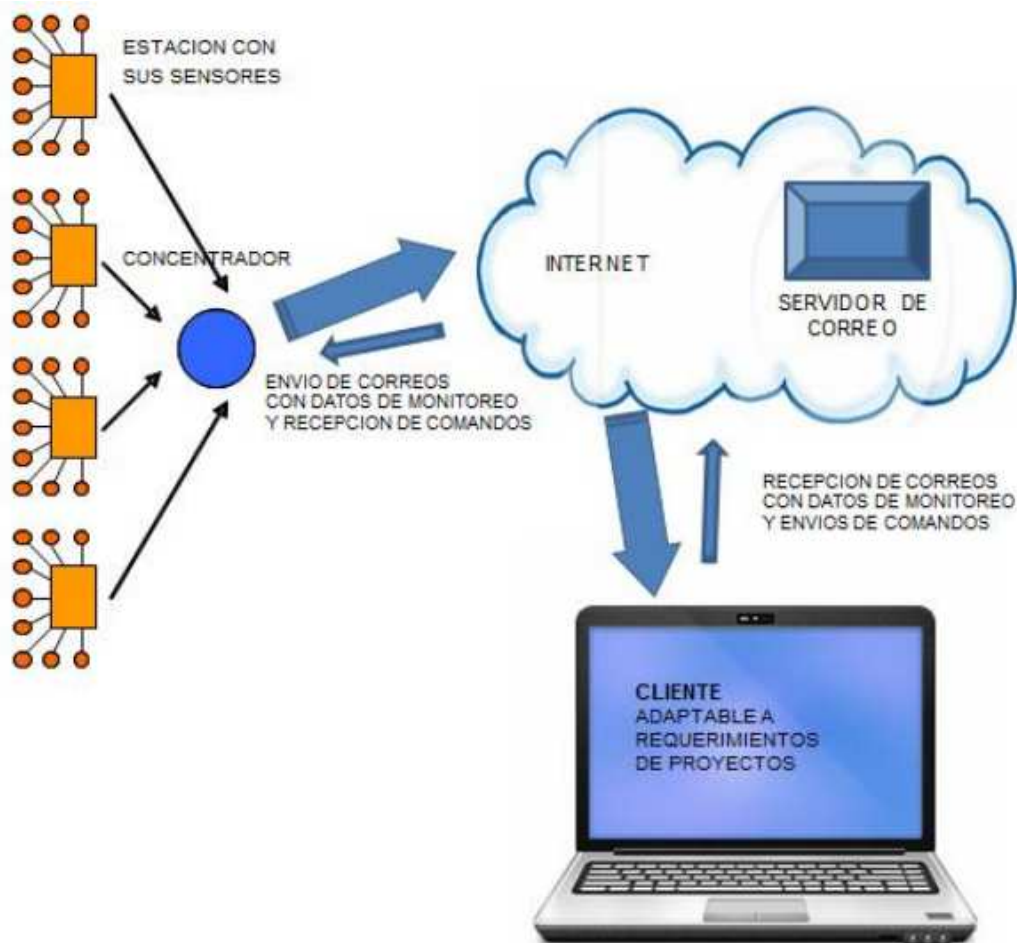


Figura N°5.2 - Esquema Servidor de Correo

Este tipo de comunicación se realiza de modo diferido entre la estación Concentradora y la aplicación "Cliente" a través de un servidor de correo electrónico. Se utiliza el asunto de los mensajes para enviar las tramas, tanto de monitoreo como de comando. La estación concentradora enviará periódicamente información de monitoreo y descargará los eventuales mensajes con tramas de comando. Los mensajes enviados con datos de monitoreo se almacenarán por tiempo indeterminado en el servidor de correo hasta que la aplicación "Cliente" los descargue, como sucede en cualquier sistema convencional de correo electrónico.

La aplicación "Cliente" enviará del mismo modo (mensajes con tramas en el asunto) comandos hacia la estación concentradora.

El protocolo se detalla en un apartado más adelante en este mismo texto.

### 5.1.3- Esquema de la Arquitectura para TCP/IP

Bosquejo de lo que se quiere implementar (Figura N°5.3):



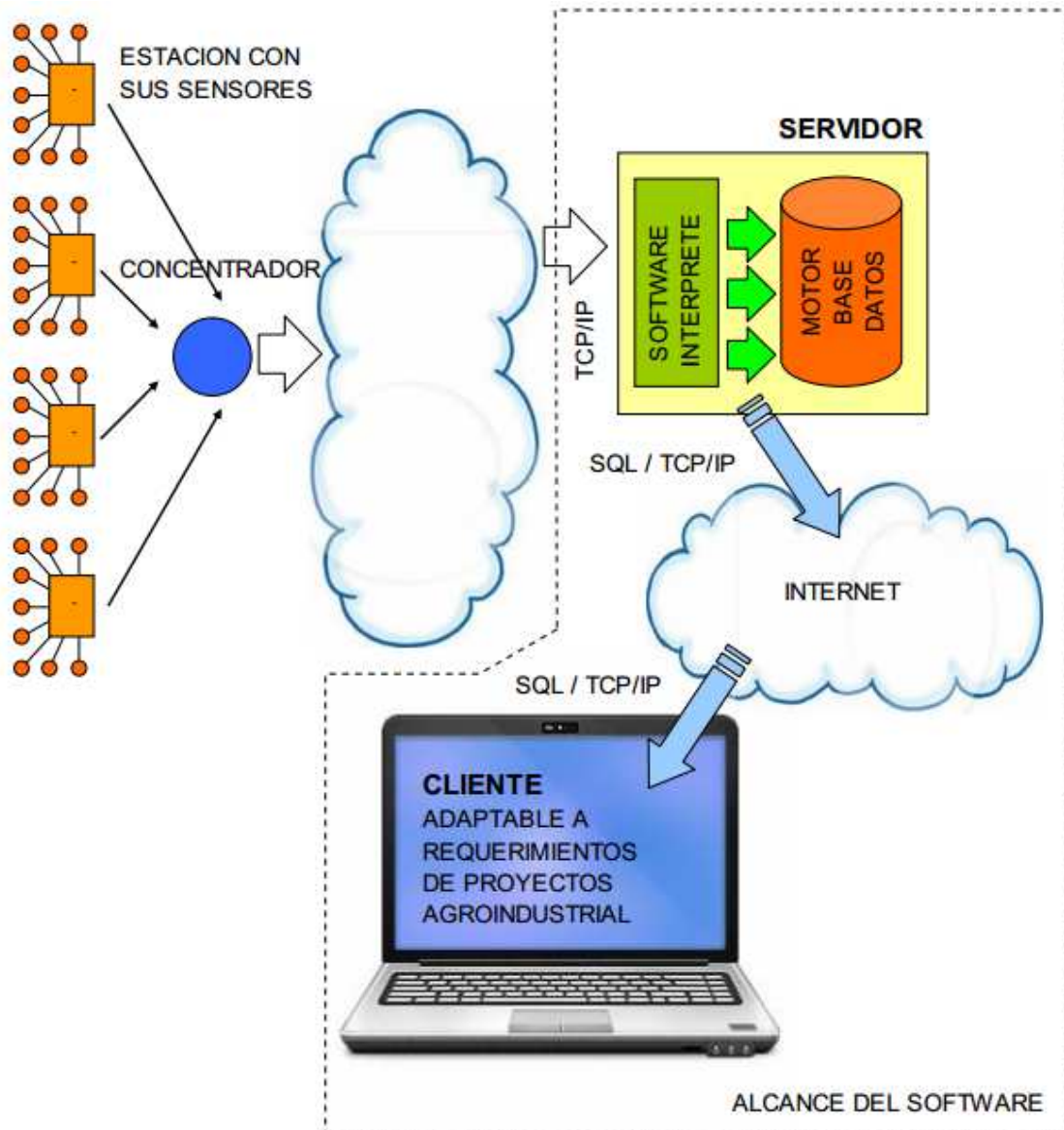


Figura N°5.3 - Esquema Arquitectura TCP/IP

Entre la estación colectora y la aplicación “Cliente” existe un “Servidor de datos”, el cual recibirá los datos de dicha estación e interpretará los datos para luego almacenarlos en una base de datos. Esta base de datos podrá ser accedida en tiempo real o a demanda a través de SQL por la aplicación “Cliente”.

Para el envío de comandos desde la aplicación “Cliente”, las tramas se retransmiten a la estación Concentradora a través del “Servidor de Datos”.

El protocolo se detalla en el siguiente apartado.

## 5.2- Tipos de Tramas

El protocolo consta de dos tipos de tramas:

- Tramas de datos
- Tramas de comandos

Todos los caracteres utilizados son caracteres ASCII.

### 5.2.1- Detalle de la Trama de datos

A continuación se detalla la trama de datos que utiliza el protocolo. Se utiliza la coma (,) para separar campos de información. Todas las tramas empiezan con un encabezado: \$D y terminan con #W

\$D,<establecimiento>,<concentrador>,<estación>,<subestación>,<fecha\_hora>,<sen/act\_1>,<sen/act\_2>,<sen/act\_3>,...,<sen/act\_9>,<sen\_act\_10>,<CSUM>,#W

Donde:

\$D	encabezado
<establecimiento>	identificador de hasta 8 caracteres, si son menos serán completados con espacios
<concentrador>	identificador de hasta 8 caracteres, si son menos serán completados con espacios
<estación>	identificador de 8 bytes/caracteres. El mismo contendrá el número de serie de la estación (64 bits provenientes de un DS2401 o equivalente) o un nombre que lo identifique (si el nombre es menor de 8 caracteres, los faltantes serán completados con espacios)
<subestacion>	identificador de 1 carácter (0-9). En caso de que la microestación posea actuadores y/o más de 10 sensores, estos serán informados como pertenecientes a diferentes subestaciones. Valores de 0 a 4 representan sensores, valores de 5 a 9 actuadores
<fechahora>	Fecha y hora del paquete, 17 caracteres. El formato de los datos es el siguiente "aa:mm:dd_hh:mm:ss"
<sen/act_n>	valor numérico del sensor o actuador. 5 caracteres, valores de sensores desde 00000-99999, valores de actuadores 00000-00100 o OONNN- OOFFF.
<CSUM>	Checksum, código de verificación de la información (2 caracteres) que representan un número hexadecimal cuyo valor es el XOR lógico de todos los bytes/caracteres entre el \$ y el CSUM sin incluirlos (se incluye la coma previa al CSUM).
#W	fin de trama

En caso de no tener todos los sensores o actuadores instalados, la trama se completa con el carácter "NNNNN" en los campos que no tienen sensor.

Ejemplo:

**\$D,SAN\_JUAN,CON\_0001,ESTAC\_01,0,14:06:30\_23:54:32,00100,00082,00024,01800,0950,00320,00028,NNNNN,NNNNN,NNNNN,46,#W**

**\$D,SAN\_JUAN,CON\_0001,ESTAC\_01,5,14:06:30\_23:54:32,00050,00025,OONNN,00030,OOFFF,OOFFF,00028,NNNNN,NNNNN,NNNNN,35,#W**

## 5.2.2- Detalle de la Trama de comando

A continuación se detalla la trama de comandos que utiliza el protocolo. Se utiliza la coma (,) para separar campos de información.

Todas las tramas empiezan con un encabezado: \$C y terminan con #W

\$C,<establecimiento>,<concentrador>,<estación>,<subestación>,<fecha\_hora>,<actuador1>,<actuador2>,<actuador3>,<actuador4>,<actuador5>,<actuador6>,<actuador7>,<actuador8>,<actuador9>,<actuador10>,<CSUM>,#W

Donde:

\$C	encabezado
<establecimiento>	identificador de hasta 8 caracteres, si son menos serán completados con espacios
<concentrador>	identificador de hasta 8 caracteres, si son menos serán completados con espacios
<estación>	identificador de 8 bytes/caracteres. El mismo contendrá el número de serie de la estación (64 bits provenientes de un DS2401 o equivalente) o un nombre que lo identifique (si el nombre es menor de 8 caracteres, los faltantes serán completados con espacios)
<subestacion>	identificador de 1 carácter (0-9). En caso de que la microestación posea actuadores y/o más de 10 sensores, estos serán informados como pertenecientes a diferentes subestaciones. Valores de 0 a 4 representan sensores, valores de 5 a 9 actuadores
<fechahora>	Fecha y hora del paquete, 17 caracteres. El formato de los datos es el siguiente "aa:mm:dd_hh:mm:ss"
<actuador...>	valor del actuador. 5 caracteres, valores de actuadores 00000-00100 o OONNN- OOFFF.
<CSUM>	Checksum, código de verificación de la información (2 caracteres) que representan un número hexadecimal cuyo valor es el XOR lógico de todos los bytes/caracteres entre el \$ y el CSUM sin incluirlos (se incluye la coma previa al CSUM).
#W	fin de trama

En caso de no tener todos los actuadores instalados, la trama se completa con el carácter "NNNNN" en los campos que no tienen sensor.

Ejemplo:

**\$C,SAN\_JUAN,CON\_0001,ESTAC\_01,5,14:06:30\_23:54:32,00050,00025,OONNN,0030,OOFFF,OOFFF,00028,NNNNN,NNNNN,NNNNN,32,#W**

## 5.3- Cliente final <C160client.exe>

### 5.3.1- Estudio de Mercado

A continuación se presentan varias alternativas de software disponibles en el mercado. El objetivo es analizar las prestaciones que brinda cada fabricante en su sistema, para ver cuales se pueden implementar en nuestro software y de esta forma poder competir con ellos.

### 5.3.1.1- Metzoo

Metzoo permite monitorear en tiempo real todas las incidencias que impacten directamente en cualquier actividad: desde servidores, hasta el tránsito de una ciudad. Esta herramienta ofrece de forma muy simple y gráfica todo lo necesario para poder visualizar, analizar y generar alertas tempranas de las variables que sean relevantes para poder comprender mejor su comportamiento. Los gráficos se actualizan en tiempo real, lo que permite tomar mejores decisiones. También se pueden armar correlaciones entre distintas variables y guardarlas en un gráfico para poder compararlas, observar relaciones de dependencia, entre otras.

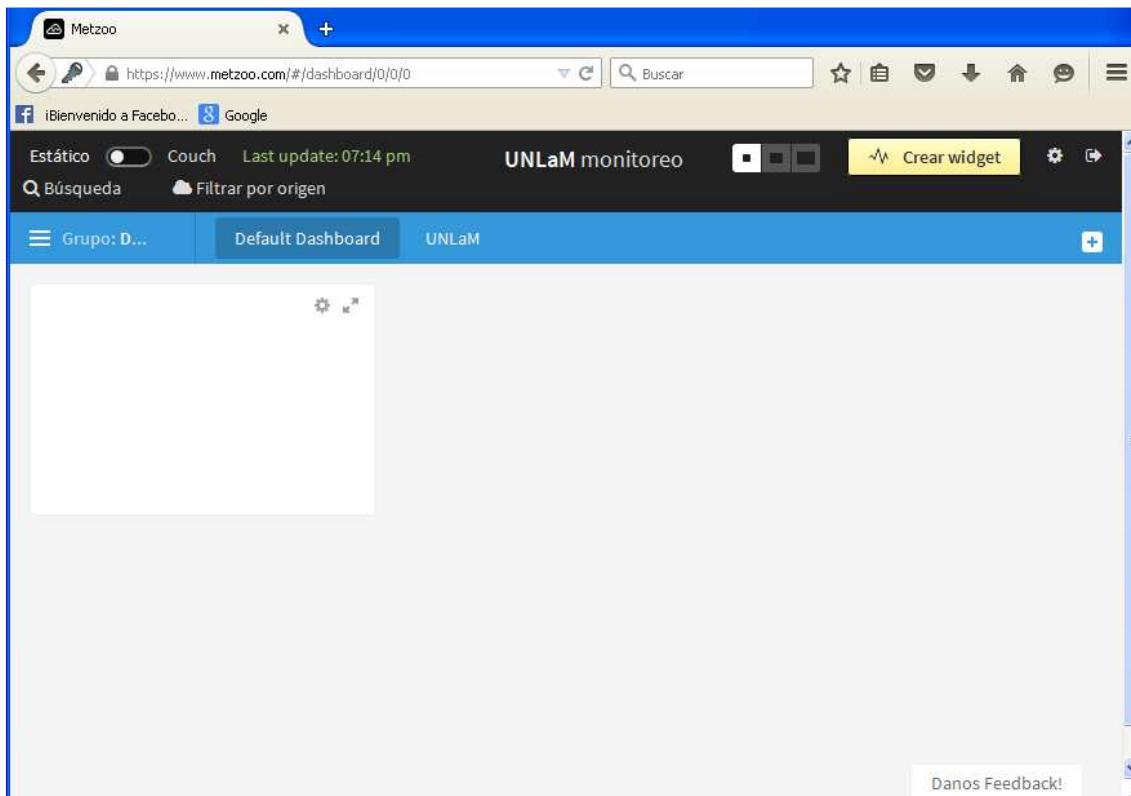
Metzoo sirve para medir desde una campaña de marketing, un proceso productivo de pequeña o gran escala, hasta el tránsito de una ciudad. Permite que un productor agrónomo controle su producción minuto a minuto, ayude a prevenir accidentes, le permita detectar anomalías y por supuesto lo impulse a tomar decisiones rápidas que eviten pérdidas económicas. Todo, sin necesidad de estar físicamente en el campo y desde cualquier dispositivo móvil.

El servicio se adquiere desde la web del fabricante. Es compatible con Mac, Windows y Linux y se integra fácilmente a Ruby, Java, Python, entre otros, con sólo unas pocas líneas de código. Una vez activo, ya sea la versión free o la versión Premium, se descarga el agente compatible con el servidor donde están almacenados los datos que se desean medir con la herramienta. Es un proceso muy sencillo, ya que son aplicaciones "one click installed". Una vez configurado, ya se puede comenzar a enviar datos. Los agentes se pueden integrar de 3 formas distintas para obtener los datos: API pública, sensores tipo Arduino y URLs.

El costo de dicha herramienta, se calcula basándose en la frecuencia con la cual se haya configurado el flujo de datos (se cobra por los datos escritos en el disco) y la información se guarda por un año. Cabe aclarar que no hay límite alguno sobre la cantidad de flujo de datos que se puede enviar [42].

Para nombrar algunos ejemplos de lo que Metzoo posibilita se puede hablar, en lo que es IoT, de la posibilidad de monitorear silos de granos, bodegas, un vivero casero (que la planta envíe un tweet cuando necesita agua), monitoreo de ritmo cardíaco, entre otras. En lo que son Smart Cities, habilita la inteligencia para administrar el tránsito, el consumo de energía de edificios públicos y luminaria, la recolección de basura, etc. Y por último, también se permite el monitoreo de Data Centres [43].

Luego de ingresar a la página del fabricante y crear un usuario, se pudo acceder a una "demo" gratis durante 30 días. La interfaz principal se puede observar en la Figura N°5.4:



**Figura N°5.4 – Software Metzoo.**

Un punto interesante que sería bueno investigar y tratar de incorporar, es la forma en que el fabricante mide los datos almacenados y calcula un costo del servicio para el usuario.

### **5.3.1.2- Booster (AgruWeather)**

Booster es una empresa argentina que busca ayudar a los pequeños agricultores del mundo a aumentar su productividad gracias a la tecnología móvil. Se estima que recibir a tiempo datos agronómicos sobre el clima y los cultivos puede facilitarle a un productor duplicar o triplicar su productividad. Según la empresa, casi el 80 por ciento de los 580 millones de agricultores del mundo tiene un acceso limitado a esta información, por lo cual pierden anualmente un 20 por ciento de sus cosechas.

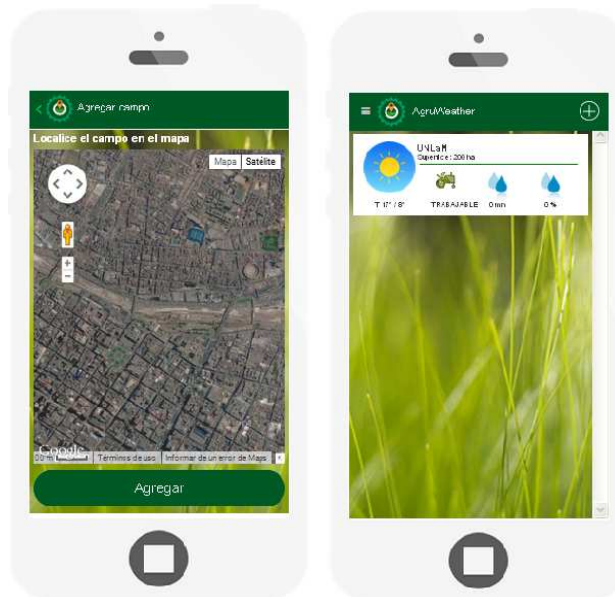
El sistema de Booster se llama AgruWeather y utiliza mensajes de texto (SMS), operadoras telefónicas y servicios de mensajería instantánea para alertar sobre posibles problemas. Básicamente el sistema recolecta múltiples fuentes de información climática, provenientes de imágenes de satélites meteorológicos de la NASA, NOAA GFS2, la Organización Meteorológica Mundial, radares y estaciones meteorológicas de todo el mundo y luego los algoritmos del software procesan estos datos para generar una predicción climática necesaria para tomar ciertas acciones [44].

AgruWeather provee información como [45]:

- Pronóstico meteorológico extendido de hasta 10 días.
- Condiciones climáticas para la estación.
- Precipitaciones (en mm.).

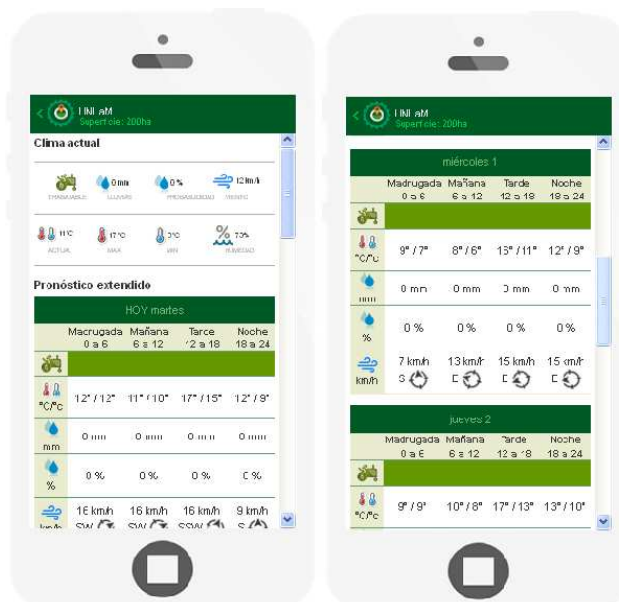
- Alertas de lluvia y granizo.

Luego de ingresar a la página del fabricante y crear un usuario, se pudo acceder a probar el software sin costo alguno. El primer paso, consiste en ubicar el campo del cual se quieren obtener los datos meteorológicos Figura N°5.5 :



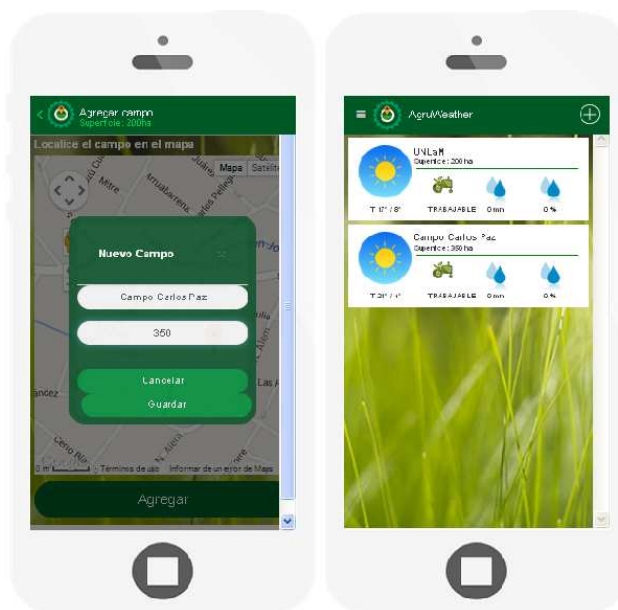
**Figura N°5.5 – Software AgruWeather(1).**

Una vez ubicado el campo, la aplicación nos da un pronóstico del clima actual y un pronóstico extendido de los próximos 4 días, tal como se puede observar en la Figura N° N°5.6 :



**Figura N°5.6 – Software AgruWeather(2).**

Dicha herramienta no sólo provee las condiciones climáticas para un solo campo, sino que permite generar un “Listado de campos” y agregar todos los campos que se deseen. En la Figura N°5.6, se puede observar el agregado de otro campo:



**Figura N°5.7 – Software AgruWeather(3).**

Luego de analizar el software, se llega a la conclusión de que sería un gran aporte para nuestro proyecto poder proveer un pronóstico actual y un pronóstico extendido para todos los campos de los distintos usuarios. Esto se podría lograr mediante un botón “Consultar pronóstico” que realice un vínculo con una página gratuita (y confiable) que provea datos meteorológicos de la ubicación del campo.

### 5.3.1.3- View Point

ViewPoint es una aplicación Web que permite rastrear, monitorear y comunicarse con activos remotos y personal agropecuarios equipados con un dispositivo de seguimiento apropiado de Honeywell Seguimiento Global. La ubicación y el estado de todos los activos pueden ser rastreados desde cualquier ordenador conectado a Internet utilizando un navegador Web [46].

View Point permite:

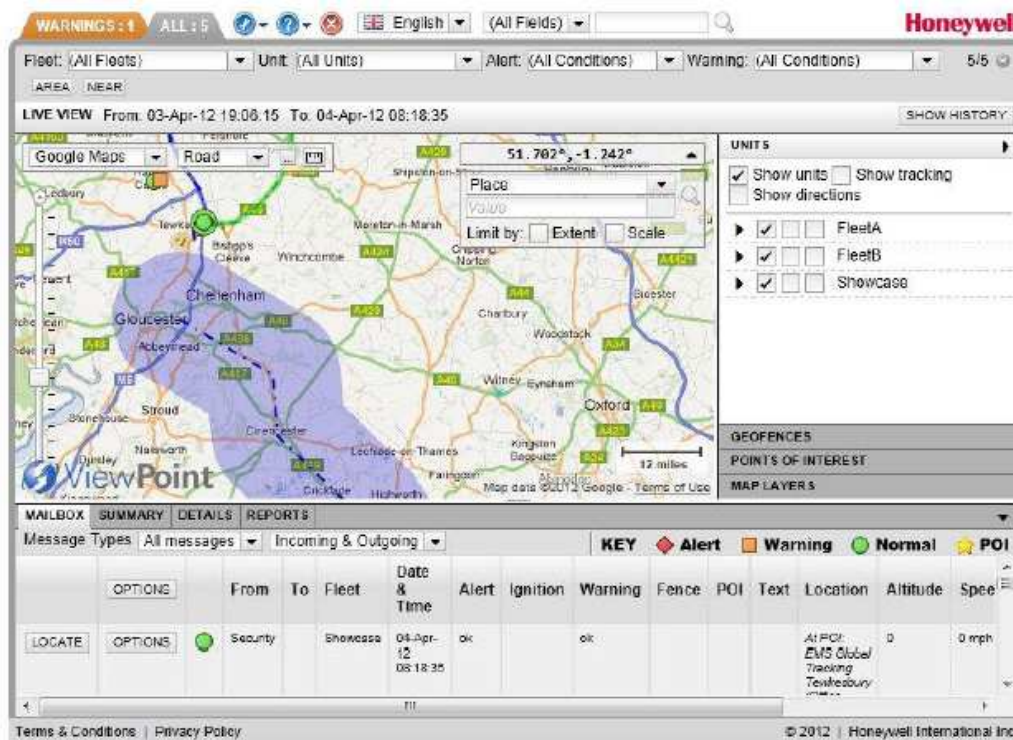
- Agregar/Eliminar establecimientos y agruparlos.
- Agregar/Eliminar usuarios, así como también asignar privilegios.
- Ver la ubicación de cada establecimiento en un mapa.
- Permite filtrar establecimientos para mostrar en el mapa según: los establecimientos que poseen una alerta o estado de alerta, los establecimientos más cercanos a un determinado lugar, entre otras.
- Ver historial de alertas.
- Generar reportes impresos.
- Aplicar filtros, para seleccionar los establecimientos que cumplan con determinada característica.
- Enviar/recibir mensajes hacia/desde los establecimientos.
- Monitorear las condiciones de alerta, así como también setear los valores de las variables que generen dichas alertas.

- Configurar alertas de notificación vía SMS o correo electrónico, en el caso de cualquier situación de alerta o de excepción (como por ejemplo cuando un operador presiona un botón de alerta o se pierde en un área peligrosa).
- Añadir puntos de interés en un mapa.

El software se compra por terminal, y dependiendo del producto, puede ser un contrato prepagado por un tiempo determinado o bien por una tarifa mensual facturada.

Para ingresar al sistema, se debe ingresar a la página Web brindada por el fabricante e iniciar sesión con el correspondiente nombre de usuario y contraseña.

En la Figura N°5.8, se puede observar la interfaz principal de dicho software:



**Figura N°5.8– Software ViewPoint.**

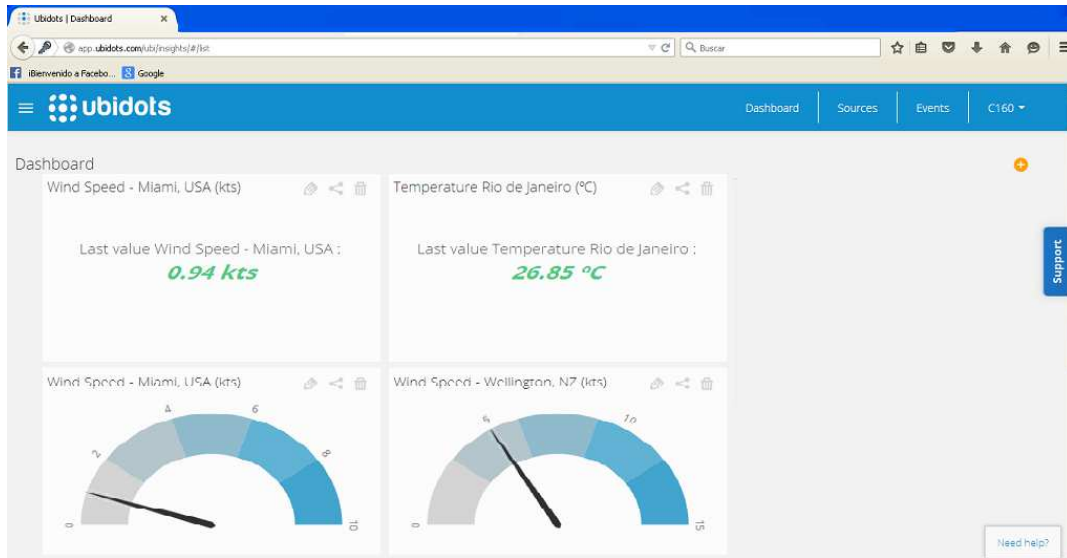
Luego del análisis del software, vemos que es uno de los más completos. Una característica interesante para implementar es el tema de notificaciones de alertas vía SMS o correo electrónico.

### 5.3.1.4- Ubidots

Ubidots tiene como objetivo procesar datos generados por sensores y dar sentido a esa información recolectada. Permite la toma de decisiones en tiempo real al analizar la información almacenada en "la nube" sobre hábitos de compra, temperatura o funcionamiento de equipos, que van desde maquinaria empresarial hasta electrodomésticos. Sus servicios de ingeniería se orientan a la salud, la alimentación y las industrias de petróleo y gas en América Latina. Ubidots ha diseñado y puesto en marcha un servicio en la nube para satisfacer la mayoría de las necesidades de proyectos de Internet de las Cosas [47].

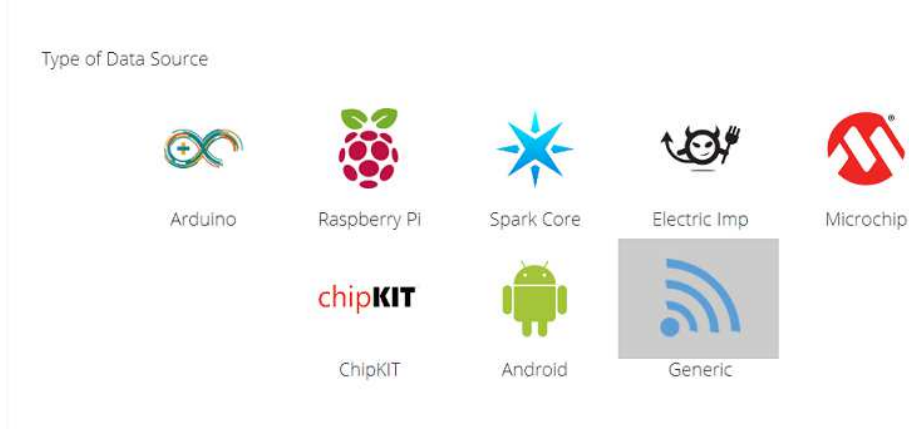


Luego de ingresar a la página del fabricante y crear un usuario, se pudo acceder a probar la herramienta sin costo alguno. En la Figura N°5.9, se puede observar la interfaz principal.



**Figura N°5.9– Software Ubidots.**

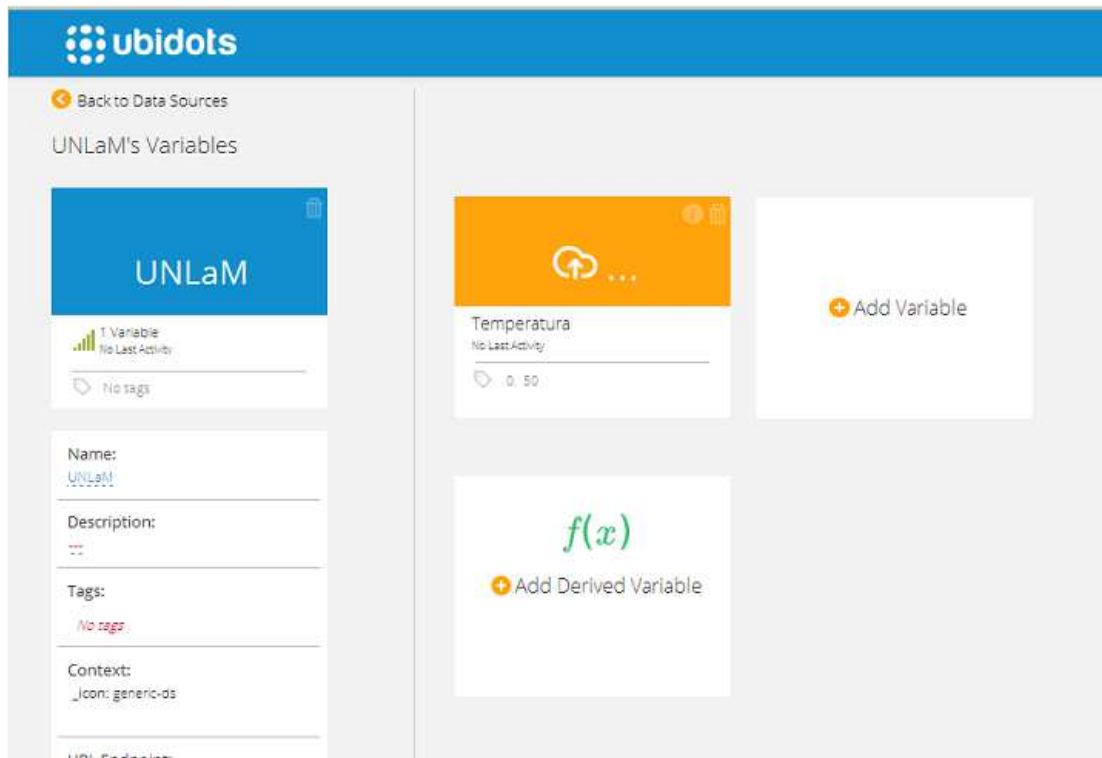
Ubidots, permite obtener los datos para procesar desde distintas fuentes, tal como se puede observar en la figura Figura N°5.10:



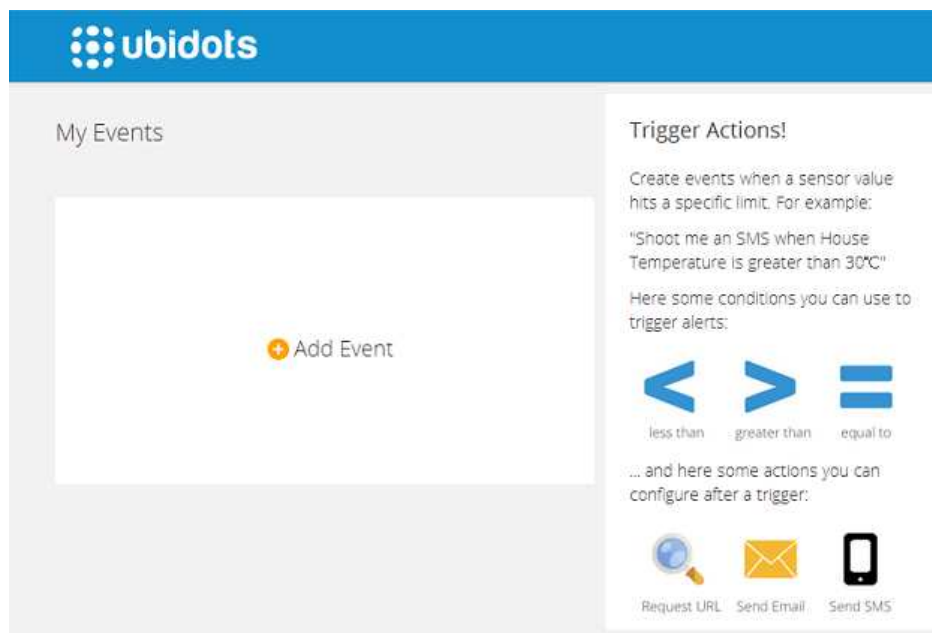
**Figura N°5.10– Software Ubidots.**

También permite crear un “Data Source” (que es el sistema del cual se obtendrán las variables a medir), así como crear variables y setear rangos de funcionamiento de las mismas. A modo de ejemplo, se creo el Data Source “UNLaM” y la variable “Temperatura”, las cuales se muestran en la Figura N°5.11.

Otra opción que ofrece el software, es crear “Eventos” (que serían alarmas) para las variables a controlar, así como enviar un SMS o E-mail en caso de que se active una de ellas, Figura N°5.12.



**Figura N°5.11– Software Ubidots.**



**Figura N°5.12– Software Ubidots.**

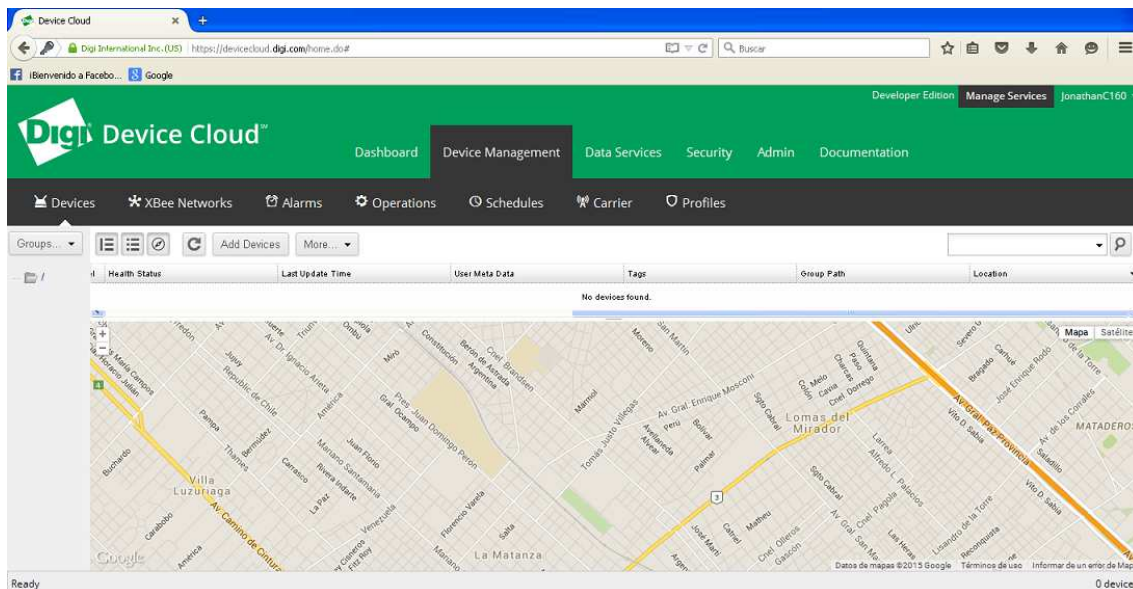
### 5.3.1.5- Digi Device Cloud

Digi es una herramienta muy similar a la anterior, que tiene como objetivo procesar datos generados por sensores y permitir la toma de decisiones en tiempo real al analizar la información almacenada en "la nube" [48].

Digi Device Cloud, permite:

- Acceder a los datos en cualquier momento y en cualquier lugar.
- Control centralizado para editar configuraciones, actualización de firmware y software de descarga.
- Supervisar el estado y ubicación de los activos remotos a través de un navegador web.
- Crear alarmas y notificaciones en base a las condiciones de datos.
- Establecer las condiciones de alarma y recibir notificación inmediata cuando un dispositivo entra en un estado en particular.
- Programar y automatizar tareas, como reinicios o actualizaciones de firmware.
- Conectar cualquier dispositivo a la nube con Digi Conector Cloud, disponible para Android, Java y dispositivos integrados.
- Oficina de Seguridad de Digi para proteger los datos de la plataforma y del dispositivo con mas de 175 diferentes controles de seguridad.
- Copia de seguridad y recuperación de datos.

Luego de ingresar a la página del fabricante y crear un usuario, se pudo acceder a una “demo” gratis. La interfaz principal se puede observar en la Figura N°5.13.



**Figura N°5.13– Software Digi Device Cloud.**

### 5.3.1.6- Telit

Telit es una herramienta de gestión web, basada en el concepto M2M (máquina a máquina), que analiza la información almacenada en "la nube". M2M es, en síntesis, la capacidad de intercambiar datos entre dos máquinas remotas, de manera tal que mediante este intercambio sea posible controlar y supervisar de forma automática procesos en los que intervienen máquinas. Para esto, todos los dispositivos a controlar, deben estar desarrollados en estrecha integración con el módulo de control de Telit [49].

Telit, ofrece:

- Administración del inventario de sus módulos de comunicación.
- Verificación de la calidad de la cobertura de red para cada unidad M2M.

- Validación de la conectividad IP para cada unidad de M2M.
- Verificación del estado del hardware de ensayo para cada unidad de M2M.
- Un ingeniero técnico de cuentas para cada cliente.
- Supervisión y respaldo técnico 24/7/365 en tiempo real.
- Monitoreo y alarma en todos los servicios.
- Gestión del módulo remoto.
- Servicio mundial de geo-posicionamiento.
- Administración de todas las conexiones desde una única plataforma sin importar el proveedor o tecnología.

Luego de ingresar a la página del fabricante y crear un usuario, se pudo acceder a una “demo” gratis. La interfaz principal se puede observar en la Figura N°5.14

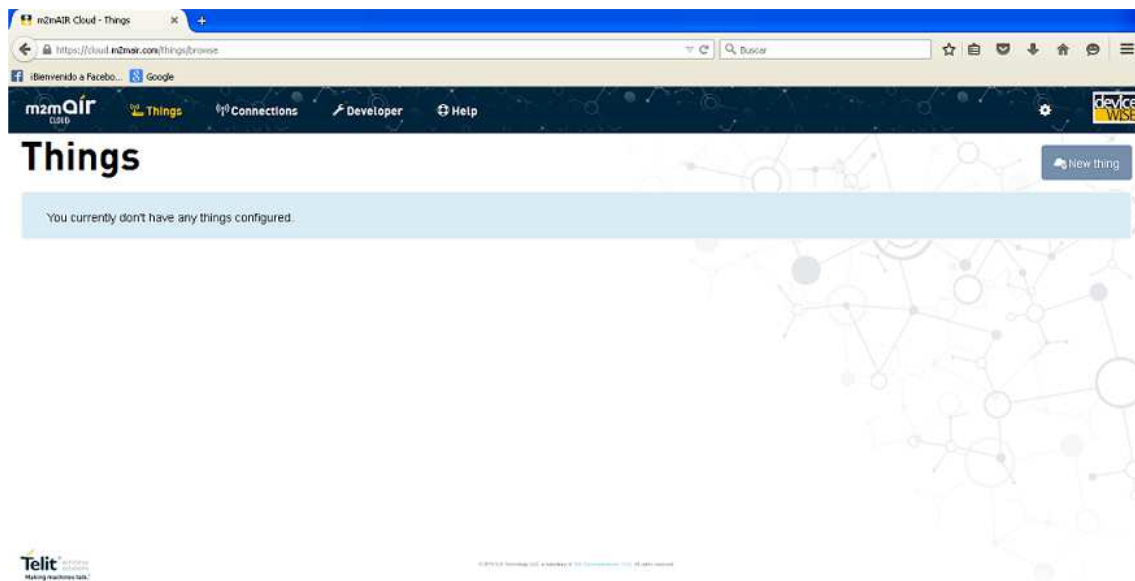


Figura N°5.14. Software Telit.

### 5.3.1.7- Resumen del Estudio de Mercado

Si bien a medida que se estudió cada software se señaló cuales propiedades eran de relevancia para nuestro proyecto, a continuación se presenta en la Tabla N°5.1 una comparación a modo de cierre del Estudio de Mercado, en el cual se resumen las principales características de cada software analizado.

Tabla N°5.1- Estudio de Mercado

Software Característica	Metzoo	Booster (AgruWeather)	View Point	Ubidots	Digi Device Cloud	Telit
<b>Costo</b>	Posee una versión Free por tiempo limitado y una versión Premium en la cual se cobra por la cantidad de datos guardados en el disco	Aplicación gratuita (disponible para Sistema Operativo Android)	El software se compra para cada terminal en la cual se lo va a utilizar y puede ser pre-pago o con factura mensual	Posee una versión Free (que permite controlar hasta 5 sensores) y diversas versiones pagas, en las cuales varia el costo según	Posee una versión Free por tiempo limitado y luego para continuar con el servicio, se debe abonar	Posee una versión Free por tiempo limitado y luego para continuar con el servicio, se debe abonar

				la cantidad de sensores a controlar		
<b>Plataforma abierta o hay que usar dispositivos de la marca</b>	Se debe descargar el agente compatible con el servidor donde se encuentran almacenados los datos a medir	No hace falta trabajar con otros dispositivos de la marca	Se debe utilizar un dispositivo de seguimiento apropiado de Honeywell	No necesita dispositivos adicionales	Se debe utilizar el conector Digi Conector Cloud, para conectar cualquier dispositivo a la nube	Los dispositivos que se deseen conectar, deben tener total integración con el módulo de control Telit
<b>Datos (gráficos, valores, etc.) en tiempo real</b>	Sí	No (los datos del clima no se actualizan constantemente)	Si	Si	Si	Si
<b>Aplicable a múltiples procesos (no sólo agropecuarios)</b>	Sí	No	No	Si	Si	Si
<b>Monitoreo desde dispositivo móvil</b>	Sí	Si (para dispositivos con SO Android)	Si (desde cualquier dispositivo con acceso a un Navegador Web)	Si (desde cualquier dispositivo con acceso a un Navegador Web)	Si (desde cualquier dispositivo con acceso a un Navegador Web)	Si (desde cualquier dispositivo con acceso a un Navegador Web)
<b>Ubicación de establecimientos en mapa</b>	No	Si	Si	Si	Si	Si
<b>Notificaciones vía SMS</b>	No	Si	Si	Si	No	No

### 5.3.2- Planteo general del software a desarrollar

Se plantea el desarrollo de un software programado en Borland C++ Builder por los motivos previamente comentados. El cual se debía conectar con un el Motor de Base de datos remotamente, mediante TCP/IP usando SQL. También tener la posibilidad de conexión directa a un concentrador por puerto serie RS232. Así como disponer de conexión a un servidor de correo electrónico donde podría adquirir información allí acumulada.

A continuación se resumen algunas de las principales funcionalidades:

- Pantalla inicial de logueo para el acceso al sistema.
- Inicio/Cierre de sesión dentro del sistema.
- Gestión de usuarios: Alta/Baja de usuarios, Modificación de contraseña y Modificación de privilegios.
- Datos Históricos: creación de un sistema de consultas para filtrar establecimientos según concentrador, estación, subestación, sensor, fecha, entre otras.
- Generación de reportes impresos en papel y PDF de dichos datos históricos.

- Módulo para geolocalización en mapa de los establecimientos con monitoreo del estado del sistema
- Generación de gráficos en pantalla de tendencias de variables, con la opción de filtrar por fecha (desde/hasta) para realizar el gráfico.
- Monitoreo en tiempo real de variables por conexión directa vía puerto serie RS232.

Según lo comentado en el punto “Estudio de Mercado”, sería de gran importancia poder incluir a futuro en nuestro software (para mejorar las prestaciones) herramientas como:

- Medición de datos almacenados y cálculo de un costo del servicio para el usuario.
- Pronóstico actual y extendido para todos los campos de los distintos usuarios.
- Notificaciones de alertas vía SMS o correo electrónico.

### **5.3.3- Comunicaciones**

El presente apartado define la manera de comunicarse por los distintos canales de comunicación disponibles en el sistema, a saber:

- Comunicación local por puerto serie RS232
- Comunicación remota utilizando un servidor de correo electrónico
- Comunicación remota por TCP/IP

#### **5.3.3.1- Comunicación local por puerto serie RS232**

Se utilizará un tasa de transferencia (baudrate) de 19200bps, 8 bit de datos, sin paridad (none) y 1 bit de parada (stop bit).

En resumen: **19200 8N1**

#### **5.3.3.2- Comunicación remota por correo electrónico**

Se deberá contar con dos cuentas de correo electrónico, una para almacenar los datos de monitoreo recibidos desde la estación concentradora y la otra donde se almacenarán los comandos enviados desde la aplicación “Cliente”.

#### **5.3.3.3- Comunicación remota por TCP/IP**

En la aplicación Server se implementan los siguientes puertos:

- Puerto 7000 TCP con polling, para la conexión al servidor a través de Ethernet.
- Puerto 7001 TCP sin polling, para la conexión al servidor a través de GPRS
- Puerto 7002 eco de lo recibido en cualquier de los otros puertos.

### **5.3.4- Descripción de la aplicación cliente.**

#### **5.3.4.1- Ingreso al sistema**

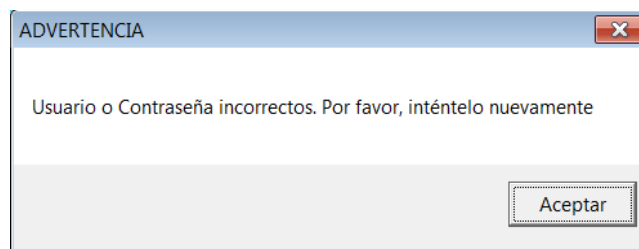
Al ejecutarse el programa, aparece la pantalla de logueo que se puede ver en la figura N° 5.15. En esta pantalla, se debe ingresar el correspondiente nombre de usuario y

contraseña para poder acceder al sistema (ambos otorgados por el administrador del mismo).



**Figura N°5.15 - Pantalla de Logeo**

En caso de que algún campo ingresado sea incorrecto, el usuario será alertado con el siguiente mensaje, figura N°5.16.



**Figura N°5.16 - Usuario o contraseña incorrectos.**

### 5.3.4.2- Pantalla principal

Una vez que se accede al sistema, la pantalla principal es mostrada, figura N°5.17.



**Figura N°5.17 – Pantalla principal.**

En dicha pantalla, se puede observar el usuario que se encuentra registrado actualmente, la versión del sistema que posee, la fecha, hora y diversos botones y menues que serán detallados a continuación.

#### 5.3.4.3- Inicio/Cierre de sesión dentro del sistema

Dentro del sistema, se podrá cerrar la sesión y volverla a iniciar (con otro nombre de usuario y contraseña, si se desea) sin necesidad de cerrar totalmente la aplicación. También se cuenta con la opción de cerrar totalmente la aplicación. Todo esto, se puede realizar desde la barra de herramientas en el menú “Sistema”, figura N°5.18.



Figura N°5.18 – Opciones menú Sistema.

#### 5.3.4.4- Establecimientos

El menú “Establecimientos”, permite ver en una tabla todos los establecimientos (con sus respectivos sensores, unidades, valores máximos y valores mínimos de dichos sensores), agregar nuevos establecimientos, eliminar establecimientos y modificar los sensores de los mismo, figura N°5.19.



Figura N°5.19 – Menú Establecimientos.

#### 5.3.4.5- Ver establecimientos y Sensores

Esta opción, permite observar todos los establecimientos (con sus respectivos concentradores, estaciones y subestaciones) y ver los sensores asociados a ellos (al igual que el nombre, unidad, valor máximo y valor mínimo de cada sensor), figura N°5.20.

#### 5.3.4.6- Agregar nuevo establecimiento

Esta pantalla permite agregar nuevos establecimientos. Para ello, hay que ingresar un nombre de establecimiento, concentrador, estación y subestación. Luego, se debe ingresar el nombre de cada sensor, al igual que su unidad de medición, valores máximos y mínimos que puede adoptar, esto se puede ver en la figura N°5.21.

#### 5.3.4.7- Borrar establecimiento

En este menú, se podrá seleccionar un establecimiento para luego eliminarlo definitivamente, figura N°5.22.



**VER ESTABLECIMIENTOS Y SENSORES**  
 Seleccione un Establecimiento:

Por favor, seleccione el establecimiento que desea observar:

ESTABLECIMIENTO	CONCENTRADOR	ESTACION	SUBESTACION	NOMBRE	UNIDAD	VALOR_MAXIMO	VALOR_MINIMO
SANLUIS	SLUIS_A	SLUIS_A_01	0	TEMP	°C	50	10
SANLUIS	SLUIS_A	SLUIS_A_01	0	HUMEDAD	%	100	10
SANLUIS	SLUIS_A	SLUIS_A_01	0	VIENTO	KM/H	190	10
SANLUIS	SLUIS_A	SLUIS_A_01	0	PRESION	ATM	100	10
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1
SANLUIS	SLUIS_A	SLUIS_A_01	0	1	1	1	1

**Figura N°5.20 – Ver establecimientos y sensores.**

**AGREGAR NUEVO ESTABLECIMIENTO**  
 Ingresar datos:

Ingreso Establecimiento:  (Máximo 10 caracteres)

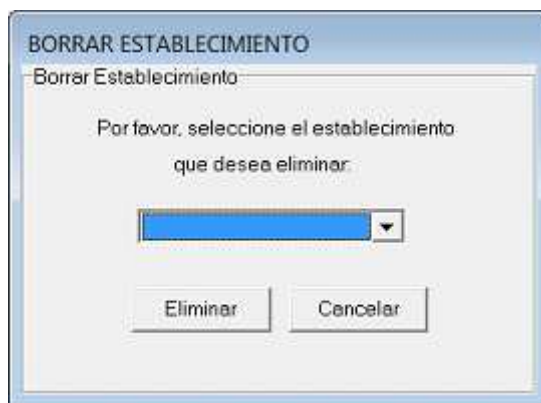
Ingreso Concentrador:  (Máximo 10 caracteres)

Ingreso Estación:  (Máximo 10 caracteres)

Ingreso Subestación:  (Máximo 10 caracteres)

Sensor	Nombre	Unidad	Valor máximo	Valor mínimo
Sensor1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sensor10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Figura N°5.21 – Agregar nuevo establecimiento.**



**Figura N°5.22 – Borrar establecimiento.**

### 5.3.4.8- Modificar Sensores

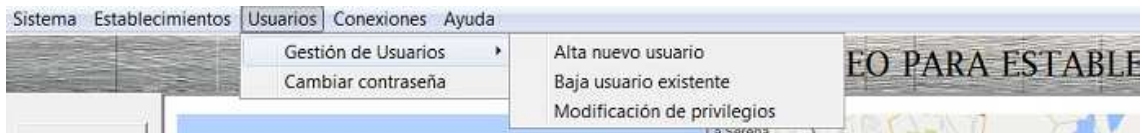
Esta opción, permite modificar todas las propiedades de los sensores asociados a un determinado establecimiento, concentrador, estación y subestación, figura N°5.23.

Sensor	Nombre	Unidad	Valor máximo	Valor mínimo
Sensor1				
Sensor2				
Sensor3				
Sensor4				
Sensor5				
Sensor6				
Sensor7				
Sensor8				
Sensor9				
Sensor10				

**Figura N°5.23 –Modificar establecimiento.**

### 5.3.4.9- Gestión de Usuarios

Desde la barra de herramientas, en el menú “Usuarios”, se podrán realizar todas las operaciones correspondientes a la “Gestión de Usuarios”. Esto comprende: Cambiar contraseña, Alta/Baja de usuarios y Modificación de privilegios. Cabe aclarar, que sólo tendrán acceso a las opciones Alta/Baja de usuarios y Modificación de privilegios, los administradores del sistema, figura N°5.24.



**Figura N°5.24 – Gestión de Usuarios.**

#### 5.3.4.10- Cambiar contraseña

En este menú, el usuario tendrá la opción de cambiar su contraseña. Para ello, primero deberá ingresar su contraseña actual (como medida de seguridad) y luego la nueva contraseña (por duplicado), figura N°5.25.

A screenshot of a dialog box titled 'CAMBIAR CONTRASEÑA'. The dialog has a title bar with a close button. The main area contains the following fields: 'Usuario Actual:' with the value 'UNLeM', 'Contraseña Actual:', 'Nueva Contraseña:', and 'Confirmar Contraseña:'. Each of the three password fields is followed by a text input box. At the bottom of the dialog are two buttons: 'Aceptar' and 'Cancelar'.

**Figura N°5.25 – Cambiar contraseña.**

En caso de que algún campo sea ingresado en forma incorrecta, el sistema mostrará el aviso correspondiente indicando cual es el campo erróneo.

#### 5.3.4.11- Alta nuevo usuario

Como ya se mencionó anteriormente, sólo tendrán acceso a este menú los administradores del sistema. En dicho menú, se podrá crear un nuevo usuario (con su correspondiente nombre, contraseña, jerarquía, etc.). La contraseña deberá ser modificada por el usuario en su primer acceso al sistema, figura N°5.26

A screenshot of a dialog box titled 'ALTA NUEVO USUARIO'. The dialog has a title bar with a close button. The main area contains the following fields: 'Nombre de Usuario:', 'Contraseña:', 'Confirmar Contraseña:', 'Jerarquía:', and 'Versión:'. Each of the three password fields is followed by a text input box. The 'Jerarquía:' and 'Versión:' fields are followed by dropdown menus with the text 'Seleccione Jerarquía' and 'Seleccione Versión' respectively. At the bottom of the dialog are two buttons: 'Aceptar' and 'Cancelar'.

**Figura N°5.26 – Alta nuevo usuario.**

En caso de que algún campo sea ingresado en forma incorrecta, el sistema mostrará el aviso correspondiente indicando cual es el campo erróneo.

#### 5.3.4.12- Baja usuario existente

En este menú, los administradores del sistema, podrán eliminar usuarios, figura N°5.27.



Figura N°5.27 – Baja usuario existente.

#### 5.3.4.13- Modificación de privilegios

Los administradores del sistema, podrán elegir un usuario y modificarle sus privilegios. En un principio, el sistema contará con una versión “Full” (en la cuál dispone de todas sus prestaciones) y una versión “Básica” (con algunas de sus prestaciones restringidas), figura N°5.28.



Figura N°5.28 – Modificación de privilegios.

#### 5.3.4.14- Conexiones

Desde la barra de herramientas, en el menú “Conexiones”, se podrán configurar las conexiones que soporta el software, figura N°5.29.

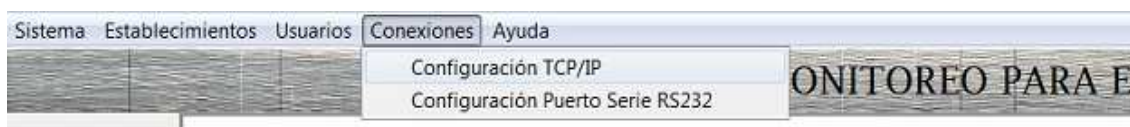


Figura N°5.29 – Conexiones.

### 5.3.4.15- Configuración TCP/IP

El motor de base de datos (Oracle, MySQL, etc.) se encarga de establecer la comunicación mediante SQL sobre TCP/IP y desde el Cliente con código SQL se accede a la información. Por ende, no es necesario establecer una comunicación TCP/IP para leer las tablas de la base de datos. Igualmente, el sistema cuenta con una conexión TCP/IP, por si llegado el caso, hubiese que realizar una conexión de este tipo desde el Cliente hacia el Servidor. Para realizar la conexión TCP/IP, se debe configurar la Dirección IP del Servidor así como el Puerto del mismo, figura N°5.30.



Figura N°5.30 – Configuración TCP/IP.

### 5.3.4.16- Configuración Puerto Serie RS232

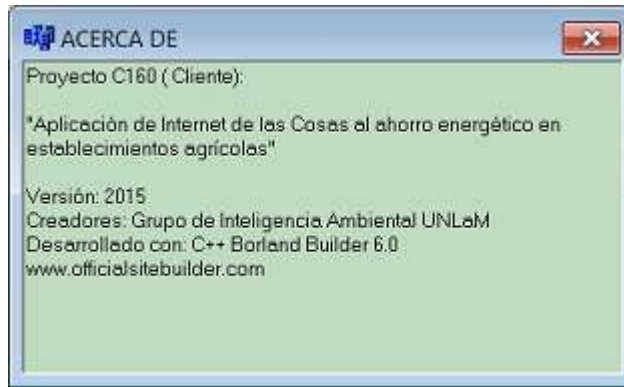
Este menú, permite realizar la selección del puerto (COM1, COM2, etc.) para la conexión Serie. Los parámetros de transferencia (velocidad, cantidad de bits de datos, paridad y bits de parada) son fijos para una conexión estándar utilizada en este tipo de sistemas, figura N°5.31.



Figura N°5.31 – Configuración Puerto Serie RS232.

### 5.3.4.17- Ayuda

Esta opción, permite acceder a la ventana “Acerca de”, la cuál contiene información del software, figura N°5.32.



**Figura N°5.32 – Acerca de.**

#### **5.3.4.18- Botones panel izquierdo Pantalla principal**

En este panel, se encuentran los botones que acceden a las principales funciones que ofrece el software, figura N°5.33.



**Figura N°5.33 – Botones Pantalla principal.**

La función de cada botón, se detalla a continuación.

#### **5.3.4.19- Ver en mapa**

Esta opción, abre una ventana que permite seleccionar el establecimiento cuya ubicación geográfica se desea observar en el mapa, figura N°5.34.



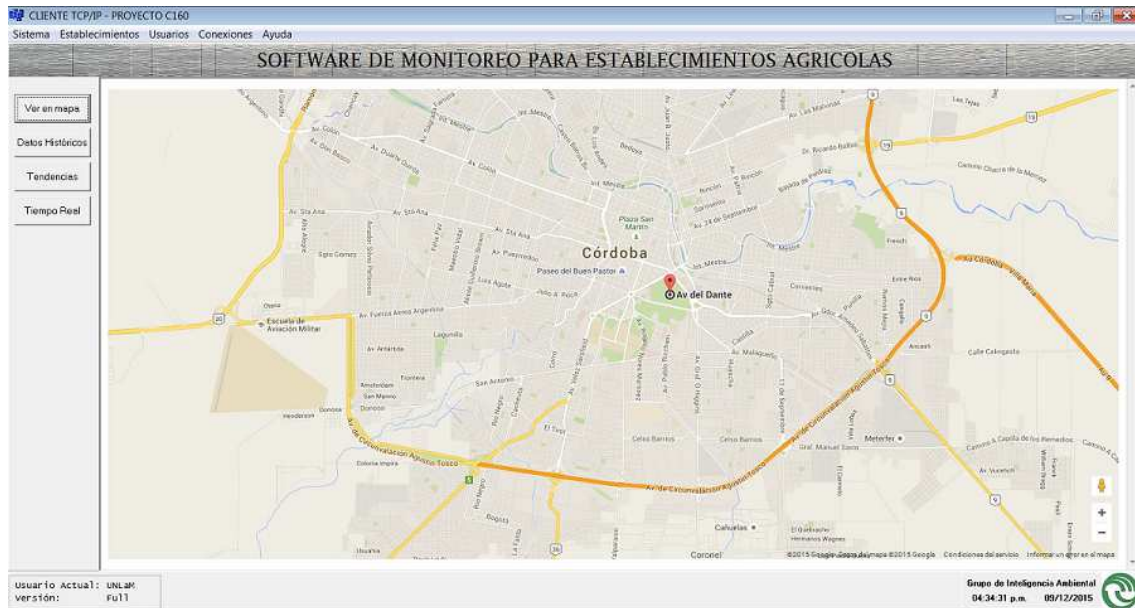
**Figura N°5.34 –Ver en mapa.**

Para realizar esto, se utilizó la aplicación Google Maps y el componente de Builder CppWebBrowser. Este componente accede a un archivo HTML que posee una línea de código provista por Google Maps (el cual llaman "código embed") que permite

observar la ubicación del establecimiento en un mapa sin que aparezcan en pantalla opciones de logeo y demás que provee el fabricante.

Actualmente se cuenta con ubicaciones genéricas, ya que esta parte del software está en desarrollo. La idea es que el usuario ingrese la ubicación del establecimiento cuando lo carga y a partir de estas coordenadas (latitud y longitud) se lo pueda ubicar en el mapa.

Si, por ejemplo, se desea observar el establecimiento “CORDOBA” la pantalla principal se mostrará de la siguiente forma, figura N°5.35



**Figura N°5.35 – Ejemplo Ubicación geográfica del establecimiento CORDOBA.**

#### 5.3.4.20- Datos Históricos

Este botón permite acceder a una pantalla en la cual se pueden visualizar los valores de los sensores de todos los establecimientos, figura N°5.36.

En el panel izquierdo, se pueden observar varias condiciones (establecimiento, concentrador, estación, subestación, sensor y fecha) para realizar filtros y poder observar los datos que nos interesen. Al tildar el primer “CheckBox”, este habilita los CheckBox de niveles inferiores para el filtrado por concentrador y por fecha. Y así sucesivamente.

El botón “Filtrar” muestra en la tabla el resultado del filtro, y el botón “Limpiar”, limpia todos los ComboBox y muestra la tabla completa (sin filtros).

El botón “Generar Reporte Impreso”, genera un archivo que se puede imprimir o exportar en PDF con los datos de la tabla (ya sea completa o con un filtro aplicado).

Esta ventana también cuenta con la opción de realizar un filtro escribiendo directamente el código SQL. Para ello, se debe tildar el CheckBox “Habilitar Modo Experto” y este habilitará el cuadro de diálogo para que se pueda escribir la sentencia SQL a ejecutar.

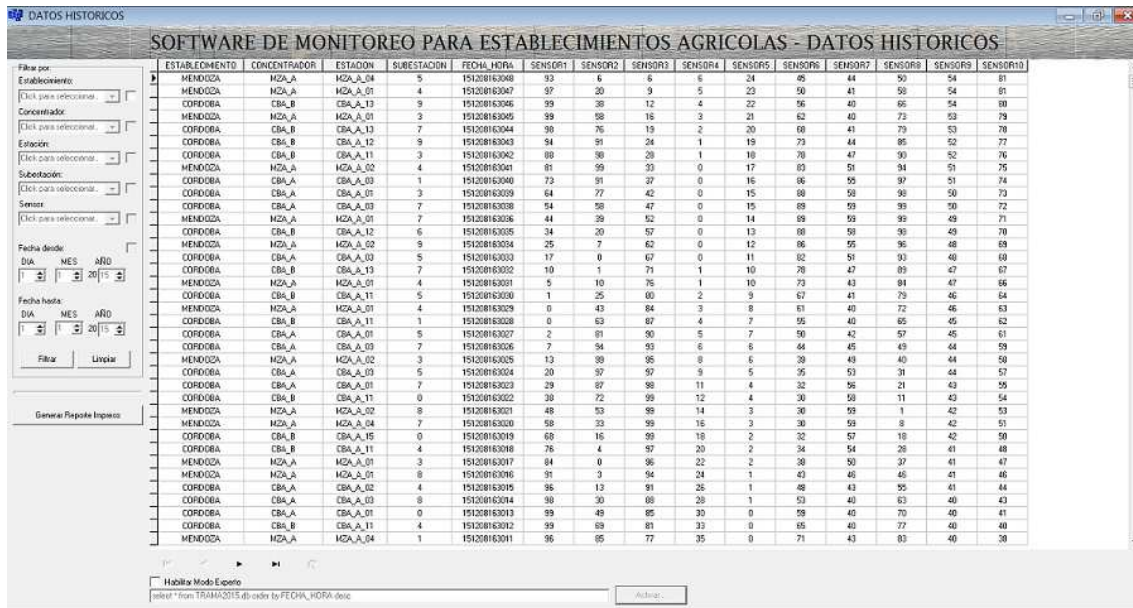


Figura N°5.36 – Datos Históricos.

A continuación, a modo de ejemplo, se realizó un filtro y luego se obtuvo el Reporte impreso correspondiente, figura N°5.37.



Figura N°5.37 – Ejemplo filtro Datos Históricos.

El Reporte Impreso obtenido, se puede ver en las figuras N°5.38 y 5.39.

### 5.3.4.21- Tendencias

El botón “Tendencias”, permite observar en forma gráfica los valores arrojados por los sensores. Para obtener el gráfico, primero se debe elegir un establecimiento, un concentrador, una estación, una subestación y un sensor determinado. El gráfico resultante, representará todos los datos históricos recibidos de ese sensor y podrá ser comparado con otro gráfico (este módulo permite visualizar en pantalla hasta 5 curvas simultáneamente). También se cuenta con la opción de filtrar por fecha, cuyo objetivo



es obtener un gráfico de los valores de un sensor dentro del rango de fechas seleccionado, figura 5.40.

Proyecto C160 - Reporte Impreso		"Aplicación de Internet de las Cosas al ahorro energético en establecimientos agrícolas"		Fecha: 10/12/2015										
				Hora: 03:49 p.m.										
Datos Históricos														
Establecimiento	Concentrador	Estación	Subestación	Fecha_Hora	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sensor9	Sensor10
MENDOZA	MZA_A	MZA_A_01	0	151208162834	86									
MENDOZA	MZA_A	MZA_A_01	0	151208162833	79									
MENDOZA	MZA_A	MZA_A_01	0	151208162804	98									
MENDOZA	MZA_A	MZA_A_01	0	151208162727	59									
MENDOZA	MZA_A	MZA_A_01	0	151208162707	81									
MENDOZA	MZA_A	MZA_A_01	0	151208162630	99									
MENDOZA	MZA_A	MZA_A_01	0	151208162304	1									
MENDOZA	MZA_A	MZA_A_01	0	151208162140	89									
MENDOZA	MZA_A	MZA_A_01	0	151208162044	81									
MENDOZA	MZA_A	MZA_A_01	0	151208162007	98									
MENDOZA	MZA_A	MZA_A_01	0	151208161745	1									

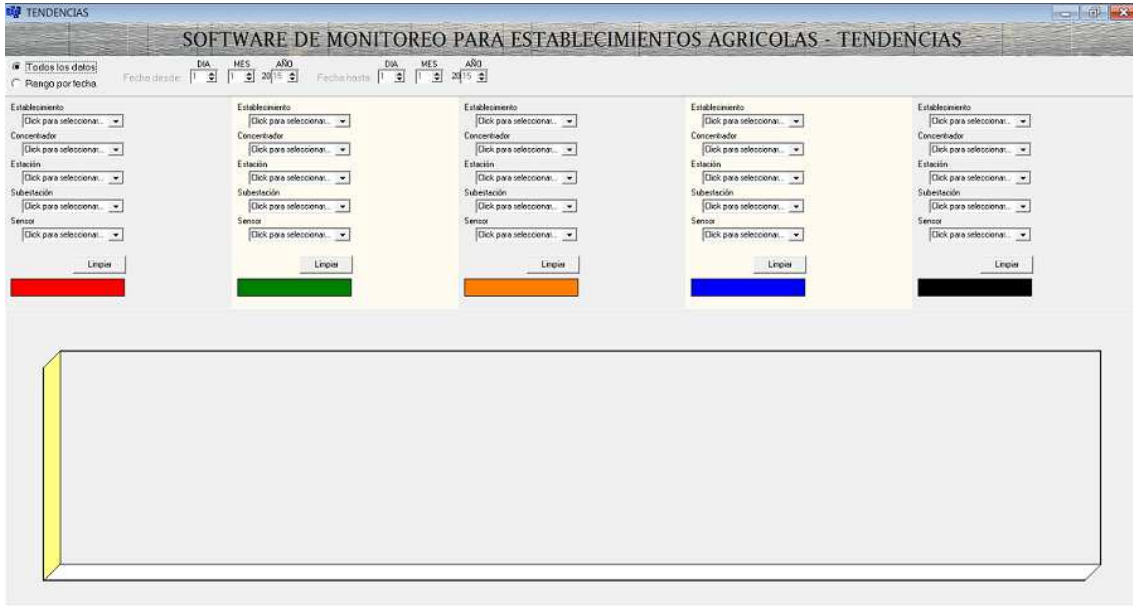
Usuario: UNLaM Página: 1

**Figura N°5.38 – Reporte Impreso filtro Datos Históricos (1<sup>ra</sup> parte).**

Proyecto C160 - Reporte Impreso		"Aplicación de Internet de las Cosas al ahorro energético en establecimientos agrícolas"		Fecha: 10/12/2015										
				Hora: 03:49 p.m.										
Establecimiento	Concentrador	Estación	Subestación	Fecha_Hora	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sensor9	Sensor10
MENDOZA	MZA_A	MZA_A_01	0	151208161743	0									
MENDOZA	MZA_A	MZA_A_01	0	151208161709	5									
MENDOZA	MZA_A	MZA_A_01	0	151208161145	52									
MENDOZA	MZA_A	MZA_A_01	0	151208160904	71									
Cantidad total de datos: 15														

Usuario: UNLaM Página: 2

**Figura N°5.39 – Reporte Impreso filtro Datos Históricos (2<sup>da</sup> parte).**



**Figura N°5.40 – Tendencias.**

Ejemplo gráfico de un sensor se puede ver en la figura N°5.41.



**Figura N°5.41 – Ejemplo Tendencias de un sensor.**

Ejemplo gráfico de dos sensores simultáneamente, se puede ver en la figura N°5.42.

#### 5.4- Servidor de datos <C160server.exe>.

##### 5.4.1- Software intérprete

Este software puede recibir información tanto desde los Concentradores asociados (datos de campo) como desde la aplicación “Cliente” (comandos). En el primer caso,

monitoreo de concentradores, recibirá tramas de monitoreo desde los concentradores mediante conexión TCP/IP, interpreta el protocolo, clasifica la información y luego almacena los datos en registros de la base de datos mediante SQL. En el otro caso, comandos a un concentrador: al recibir una trama de comando desde el “Cliente”, esta será retransmitida al concentrador.

Se programó en Borland C++ Builder 6.0 por disponer de las licencias de uso en el ámbito de la Universidad Nacional de La Matanza.

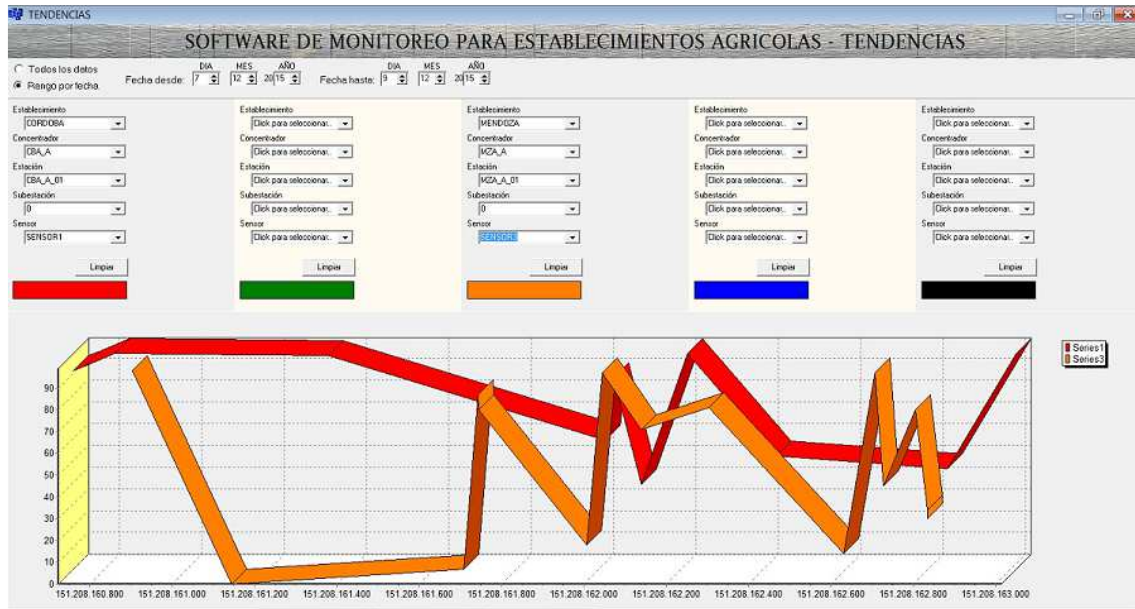


Figura N°5.42 – Ejemplo Tendencias de dos sensores.

A continuación se muestra la GUI (Interfaz gráfica de usuario) del software servidor (Figura N° 5.43)

ESTABLECIMIENTO	CONCENTRADOR	ESTACION	SUBESTACION	FECHA_HORA	SENSOR1	SENSOR2	SENSOR3	SENSOR4	SENSOR5	SENSOR6	SENSOR7	SENSOR8	SENSOR9	SENSOR10	CheckSum
MENDOZA	MZA_A	MZA_A_04	7		59	69	54	52	51	89	59	59	99		
CORDOBA	CBA_B	CBA_A_14	4		69	85	59	54	52	87	56	98			
MENDOZA	MZA_A	MZA_A_04	6		78	96	64	57	53	84	53	95			
CORDOBA	CBA_B	CBA_A_14	3		85	99	69	59	54	90	49	92			
CORDOBA	CBA_A	CBA_A_02	9		15	99	3	77	97	98	50	37			
MENDOZA	MZA_A	MZA_A_02	7		8	96	5	75	98	43	46	46			
MENDOZA	MZA_A	MZA_A_04	8		3	85	8	73	98	48	42	55			
CORDOBA	CBA_B	CBA_A_15	0		0	69	11	71	98	54	40	63			
CORDOBA	CBA_A	CBA_A_03	8		0	57	86	4	91	57	40	67			
CORDOBA	CBA_B	CBA_A_12	0		0	37	83	3	91	63	40	74			
MENDOZA	MZA_A	MZA_A_01	7		2	20	79	2	90	69	41	81			
CORDOBA	CBA_B	CBA_A_14	7		6	6	75	1	89	74	44	86			
MENDOZA	MZA_A	MZA_A_03	8		12	0	70	1	88	79	48	91			
MENDOZA	MZA_A	MZA_A_03	0		19	1	65	0	88	83	52	94			
MENDOZA	MZA_A	MZA_A_02	5		28	10	61	0	87	87	56	97			
MENDOZA	MZA_A	MZA_A_02	4		37	26	56	0	86	89	58	99			
MENDOZA	MZA_A	MZA_A_03	4		50	50	50	50	50	90	60	100			
MENDOZA	MZA_A	MZA_A_02	5		59	69	54	52	51	89	59	99			
CORDOBA	CBA_A	CBA_A_03	6		69	85	59	54	52	87	56	98			
MENDOZA	MZA_A	MZA_A_03	6		78	96	64	57	53	84	53	95			
MENDOZA	MZA_A	MZA_A_01	1		85	99	69	59	54	80	49	92			
CORDOBA	CBA_A	CBA_A_02	3		69	85	59	54	52	87	56	98			
MENDOZA	MZA_A	MZA_A_02	6		78	96	64	57	53	84	53	95			
CORDOBA	CBA_A	CBA_A_02	0		85	99	69	59	54	80	49	92			
MENDOZA	MZA_A	MZA_A_01	7		92	95	73	62	56	76	45	87			
MENDOZA	MZA_A	MZA_A_02	1		96	83	78	64	57	70	42	82			

Figura N° 5.43. Interfaz Gráfica del usuario

La aplicación dispone varios puertos para realizar su trabajo:

- Puerto con Pooling, una vez abierto envía una trama de datos al cliente periódicamente, la cual se puede emplear para interrogar al cliente.
- Puerto son Pooling, una vez abierto solo queda en escucha de paquetes enviados desde la aplicación cliente.
- Puerto Replicador, una vez abierto dicho puerto realizara un eco de los datos recibidos en lo otros puertos abiertos de la aplicación.

#### **5.4.2- Motor de base de datos**

Se trata de un software comercial de licencia gratuita, en lo posible libre, que se instalará, configurará y utilizará. Corre, en general, como un servicio y permite ser consultado mediante SQL, local y remotamente, de varias maneras incluyendo seguridad y control de usuarios.

Podría ser MySQL, SQLite, MSSQLServer, entre otros. A continuación vamos a presentar algunos de ellos, pero previamente aclararemos algunos términos.

Base de datos relacionales: Son aquellas que Permiten establecer interconexiones o relaciones entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas.

ODBC (Open Database Connectivity): Estándar de acceso a Bases de Datos desarrollado por Microsoft cuyo objetivo es hacer posible el acceso a cualquier aplicación, sin importar que Sistema Gestor de Base de Datos (DBMS por sus siglas en ingles) almacene los datos.

Licenciamiento dual: Este tipo de licenciamiento es común en el caso del software, donde un producto software libre (que implica que los trabajos derivados deben ser también libres) puede ser licenciado a empresas para permitir su uso comercial de una forma no libre.

##### **5.4.2.1- Paradox y BDE (Borland Database Engine)**

Es un sistema de base de datos relacional de fácil uso que permite a los usuarios guardar y recuperar datos, crear formularios profesionales e informes en que se despliegan los datos. Desarrollado por el Ansa Software en 1985, comprado por Borland International Inc. en 1987, y a través de una alianza anterior entre Borland y Corel en 1996, Borland autorizó Paradox a Corel. En el año 1997 Corel actualizo completamente este sistema de base de datos. Tuvo una cuota del 85% del mercado de las bases de datos de escritorio hasta el año 1995 que Microsoft incluyó Access en el paquete Microsoft Office Professional.

Entre sus características principales podemos citar:

- Puede funcionar satisfactoriamente en ambientes Cliente-Servidor locales o hasta de acceso remoto si se le usa en una red WAN.
- Habilita diferentes ODBC, lo que le permite cambiar sus tablas de datos con cualquier sistema de gestión de base de datos de relacionales de base compatible, como Oracle, SQL, Microsoft Access, etc.
- Trae incorporado el intérprete de SQL, llamado Borland's Local SQL, que es una parte del estándar ANSI-92 SQL, del año 1992.

- Paradoja proporcionan una gran flexibilidad con el tamaño de su base de datos, puede crear hasta dos mil millones de registros por tabla y hasta 255 campos por registro.

Si bien estas características fueron importantes al momento de su lanzamiento, actualmente se encuentran ampliamente superadas, al punto que es un producto discontinuado. Sumado a esto, no es un software gratuito y solo puede ser utilizado en sistemas operativos Windows. Sin embargo se ha tenido en cuenta porque la misma permite una rápida implementación y además está integrada en el paquete de Borland C++, el cual es el utilizado en la cátedra de Informática avanzada y del cual se poseen licencias [120][121].

#### **5.4.2.2- MySQL**

MySQL es un sistema de bases de datos relacional, multihilo y multiusuario, desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009, desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google, entre otros.

Entre sus características principales podemos citar:

- Es un software Libre.
- Puede funcionar con una gran variedad de sistemas operativos.
- MySQL soporta todas las características clave de bases de datos relacionales, entre las que podemos nombrar SQL-92 y SQL-99, Drivers para ODBC, JDBC, .NET y C++, entre muchas otras más.
- El tamaño de la tabla efectiva máxima para bases de datos MySQL es generalmente determinada por las limitaciones del sistema operativo en el tamaño de los archivos, no por los límites internos de MySQL.

Este software presenta características muy atractivas al momento del estudio, principalmente el tipo de licenciamiento que este posee. La única desventaja que se podría citar es que es producto más complejo de instalar, implementar y desplegar, por lo menos con respecto a Paradox [122][123][124].

#### **5.4.2.3- Oracle**

Oracle Corporation es una compañía de software fundada en 1977 que desarrolla bases de datos (Oracle Database) y sistemas de gestión de bases de datos. Cuenta además, con herramientas propias de desarrollo para realizar aplicaciones, como Oracle Designer, Oracle JDeveloper y Oracle Developer Suite. Desarrolla Oracle Database, desde hace más de 30 años, que es un sistema de gestión de base de datos de tipo objeto-relacional. Se considera a Oracle Database como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad, y soporte multiplataforma. Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia la oferta de otros sistemas de administración de bases de datos con licencia libre como el MySQL.

Entre sus características principales podemos citar [125][126][127][128]:

- Puede funcionar con una gran variedad de sistemas operativos.

- Soporta gran variedad de los tipos de datos relacionales estándares, así como también datos nativos como XML, texto, imágenes, documentos, audio, y datos espaciales. El acceso a la información es realizado a través de interfaces estándares como SQL, JDBC, SQLJ, ODBC.Net, OLE.Net y ODP.Net, SQL/XML, XQuery y WebDAV.
- El tamaño máximo de los archivos de base de datos depende del sistema operativo.
- Limitado por el tamaño máximo de archivo del sistema operativo.

Comparativas más detalladas sobre características y desempeño de este software en comparación con el MySQL se puede encontrar en las siguientes referencias [129][130]. Si bien deben existir un gran número de diferencias entre Oracle y MySQL, a nuestros fines prácticos del proyecto ambos son igual de útiles, teniendo Oracle como desventaja el tema del licenciamiento, el cual no es gratis, y un nivel mayor de dificultad respecto a la instalación, la implementación y el despliegue.

#### **5.4.2.4- Selección de motor de base de datos**

El proyecto requiere una base de datos relacional que funcione con la arquitectura Cliente Servidor, y que estos puedan estar conectados por internet geográficamente distanciados. Características las cuales podemos encontrar tanto en MySQL como en Oracle. Se eligió Oracle Database como motor de base de datos porque es el sistema para manejo de datos más seguro, confiable, probado y con soporte del mercado. Adicionalmente, la empresa Oracle Corporation lanzó a principio del año 2014 una versión nueva con licencia libre.

Es importante destacar que las consultas se realizarían a través de SQL, lo que permitiría una migración de un motor a otro sin mayor dificultades.

#### **5.4.3- Implementación de base de datos.**

##### **5.4.3.1- Oracle**

Tal como se planteó la sección anterior se seleccionó Oracle para comenzar el desarrollo. A continuación se explicará el modelo de base de datos implementado y los pasos realizados.

##### **5.4.3.1.1- Modelo de base de datos**

##### **5.4.3.1.2- Pasos a realizar**

##### **5.4.3.1.3- Conexión a Base de Datos mediante ODBC**

A continuación se detallan los pasos a seguir para generar la conexión al motor de base de datos.

1. En **Herramientas Administrativas** del **Panel de Control** se identifica el siguiente icono para poder comenzar con la configuración (Figura N°5.44) .

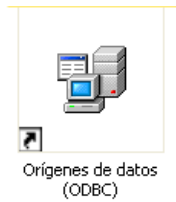


Figura N°5.44 – Icono Herramienta Administrativa

2. Creamos un DSN (DataSource Name) en el sistema, seleccionando la solapa “**DSN de sistema**” y presionamos el botón “**Agregar**” (Figura N°5.45).

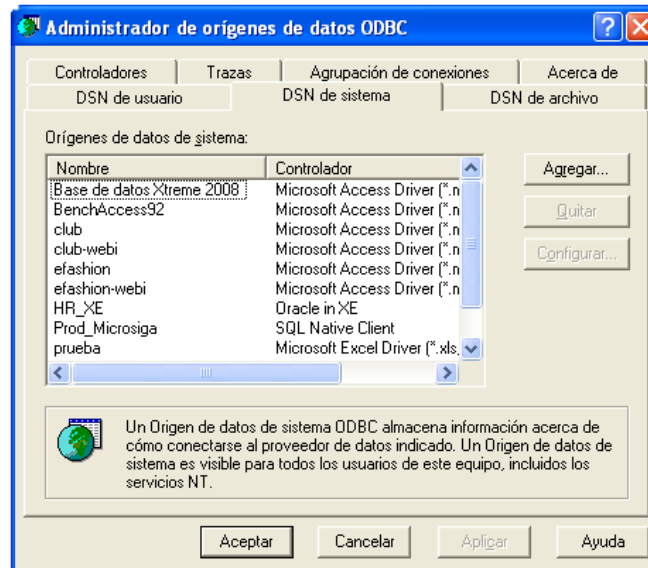


Figura N° 5.45 – Solapa DSN de sistema

3. Seleccionamos como nuevo origen de datos **Oracle in XE** y presionamos el botón **Finalizar** (Figura N°5.47).

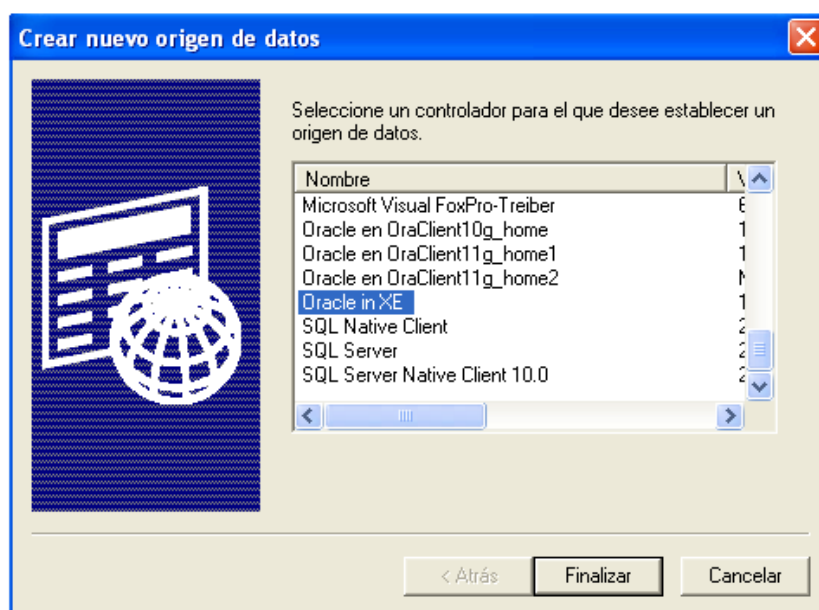
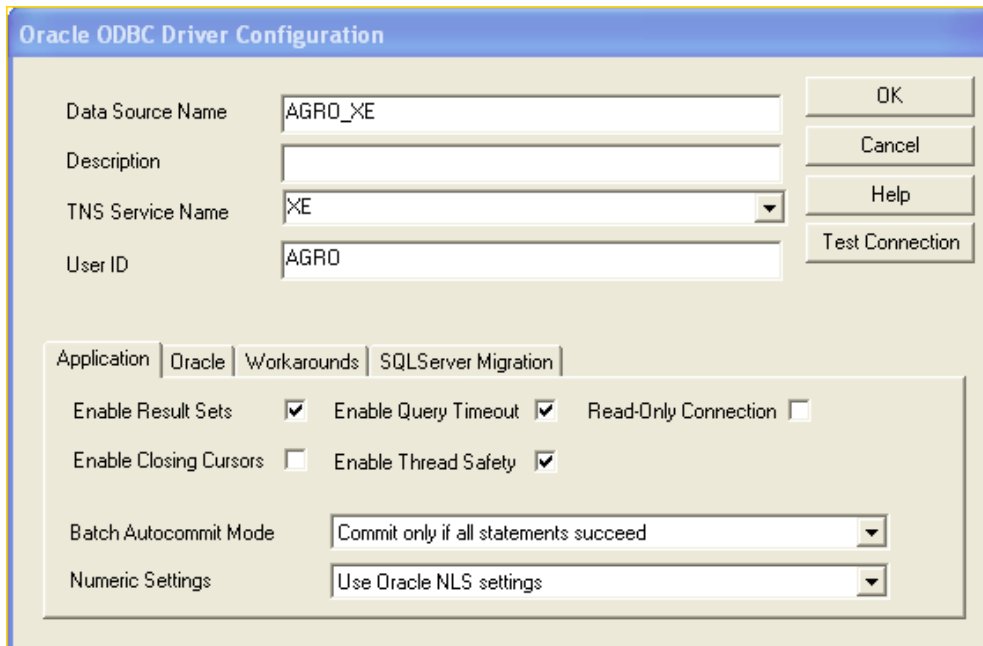


Figura N° 5.47- Pantalla Nuevo origen de datos

4. Se completan las características de conexión.
5. Data Source Name: Nombre del definido por nosotros que luego haremos referencia en los controles de conexión del C++ Builder: **AGRO\_XE**.
6. TNS Service Name: **XE**. (Nombre de la base de datos).
7. User ID: El usuario de conexión a la base de datos, **AGRO** (Figura N°5.48).



**Figura N°5.48 – Pantalla configuración usuario**

8. Probamos conexión con el botón **Test Connection**, e introducimos la contraseña del usuario **AGRO**. Y por último se pulsa el botón **OK** como se ve en la figura próxima.

**IMPORTANTE:** Contraseña (*password*) del usuario de **AGRO** es también **agro2014** (Figura 5.49).

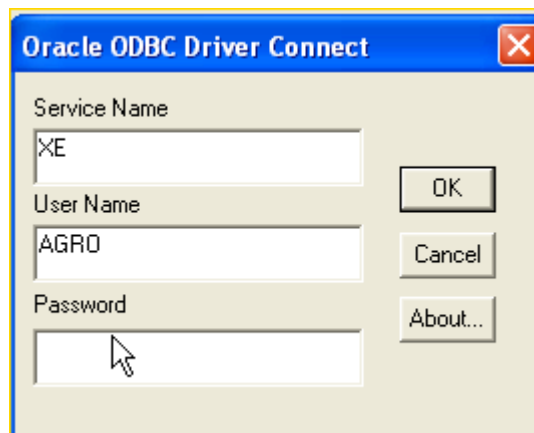






Figura N°5.49 –Pantallas de conexión.

#### 5.4.3.1.4- Conexión desde el proyecto software con base de datos

Para la conexión a la base de datos llamada XE, base de datos de muestra distribuida con el software Oracle 11G Express Edition. Se realizan los siguientes pasos.

Creación de ODBC que genere la conexión a la base de datos XE.

Colocar el componente **TDatabase** , el cual se conectará al ODBC creado.

El componente (TDatabase) **Database1** tendrá un nombre: en este caso será **"local"** (Figura N°5.50).

El alias\_name definido es el DSN creado llamado AGRO\_XE.  
Se introduce en parameter overrides. Password=agro2014.

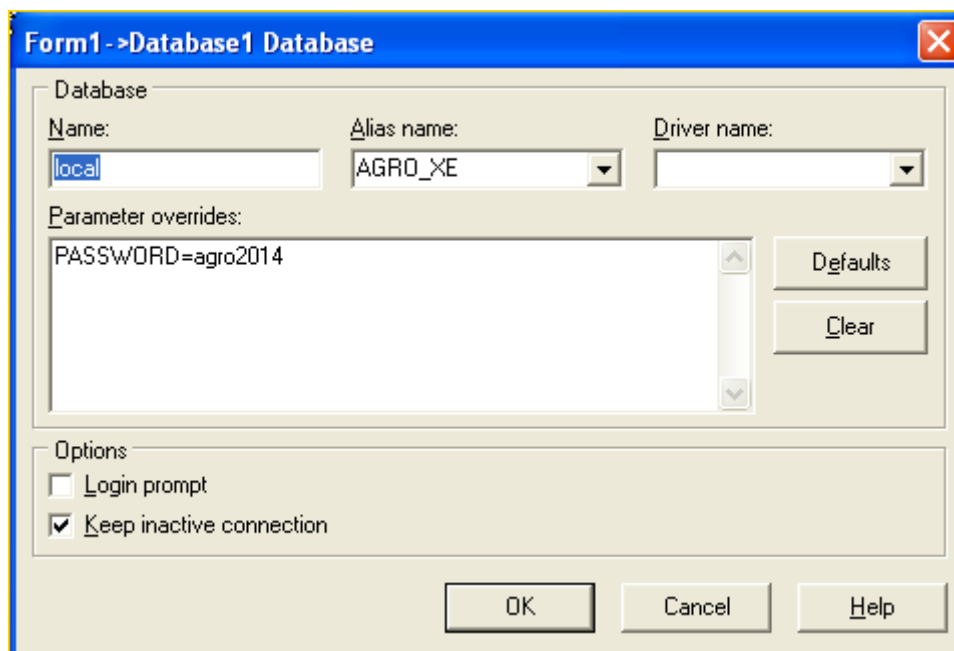
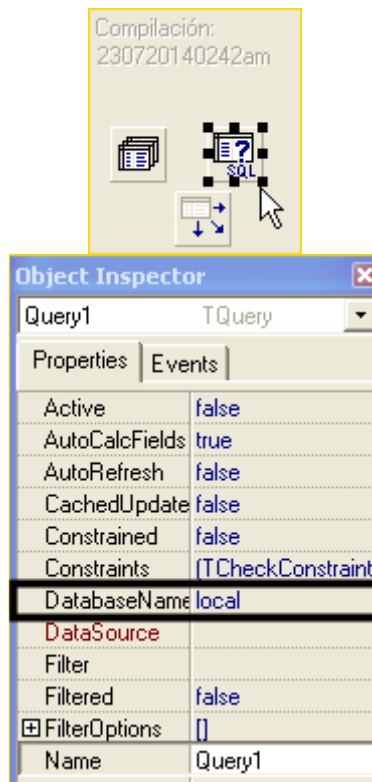


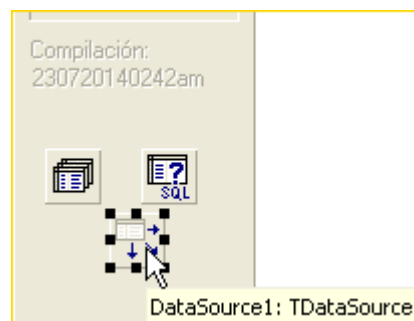
Imagen N°5.50-Pantalla Database

En el componente (TQuery) **Query1** utilizado se configura la propiedad DatabaseName **"local"**, que hace referencia al componente (TDatabase) **Database1**.



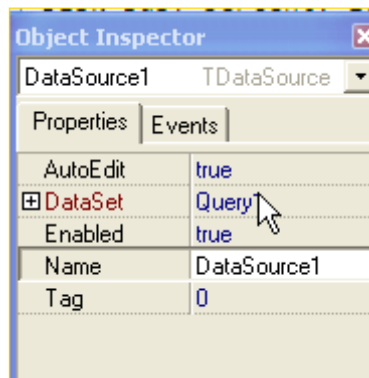
**Figura N°5.51 - Pantalla Database propiedades**

Posteriormente en un componente (TDataSource) **DataSource1**, se configurará la propiedad Dataset con el nombre de la componente TQuery “**Query1**”



**Figura N°5.52 - Pantalla Database propiedades**

En el componente (TDBGrid) **DBGrid1** se configurará la propiedad DataSource con el nombre del componente TDataSource “**DataSource1**”, resultando en pantalla cuando se ejecute el software:



**Figura N°5.53 - Pantalla Database Propiedades**

#### 5.4.3.1.5- Modelo de consulta SQL básica

Se lista a continuación el código para realizar la transacción en la base de datos.

```
/*-----
insert sobre la tabla AGRO.TRAMA la cual registra toda la información que se recibe
de los sensores.
Los datos se cargarán en labels y esos valores se insertarán sobre las tablas
-----*/
```

```
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("insert into AGRO.TRAMA");
Query1->SQL-
>Add("(ESTABLECIMIENTO,CONCENTRADOR,ESTACION,SUBESTACION,SENSO
R1,SENSOR2,SENSOR3,SENSOR4, SENSOR5, SENSOR6,SENSOR7, SENSOR8,
SENSOR9, SENSOR10 )");
Query1->SQL->Add(" values('"+ESTABLECIMIENTO->Caption+"', '"
+CONCENTRADOR->Caption+"', '"+ESTACION->Caption+"', '"
+SUBESTACION->Caption+"', '"+SENSOR1->Caption+"', '"
+SENSOR2->Caption+"', '"+SENSOR3->Caption+"', '"
+SENSOR4->Caption+"', '"+SENSOR5->Caption+"', '"
+SENSOR6->Caption+"', '"+SENSOR7->Caption+"', '"
+SENSOR8->Caption+"', '"+SENSOR9->Caption+"', '"
+SENSOR10->Caption+"' )");
Query1->ExecSQL();
```

#### 5.4.3.1.6- Cambio de motor de base de datos

Si bien la elección de Oracle Database suponía la elección adecuada para lograr los objetivos planteados por el proyecto, los cuales involucraban gran cantidad de registros de información de múltiples de establecimientos funcionando en simultáneo. Se nos presentó un problema al comenzar a trabajar más de un miembro del grupo en esta parte del proyecto, el mismo estaba asociado a la instalación de Oracle Database, más el despliegue de nuestro software. Esto presentaba una complejidad elevada y variable según el sistema operativo en que se instalara, lo que dificultaba el trabajo en grupo.

Por lo tanto se planteó realizar un cambio, teniendo en cuenta que durante el desarrollo no usaríamos gran número de registros sino tan solo algunas centenas para

poder hacer las pruebas y testing del software. Se propuso desarrollar con una alternativa de motor de base de datos más liviano, más sencillo de instalar y configurar y fundamentalmente más fácil de desplegar en las computadoras donde se implementaría y otras donde se probaría, con la condición que el software a realizar sea 100% compatible luego para poder desconectarse de esta base de datos reducida y conectarse con la base de datos definitiva en Oracle Database.

A partir de ahí se desarrollaron ambas interfaces de usuario: SERVIDOR y CLIENTE usando el BDE (Borland Database Engine) y Paradox, también de Borland.

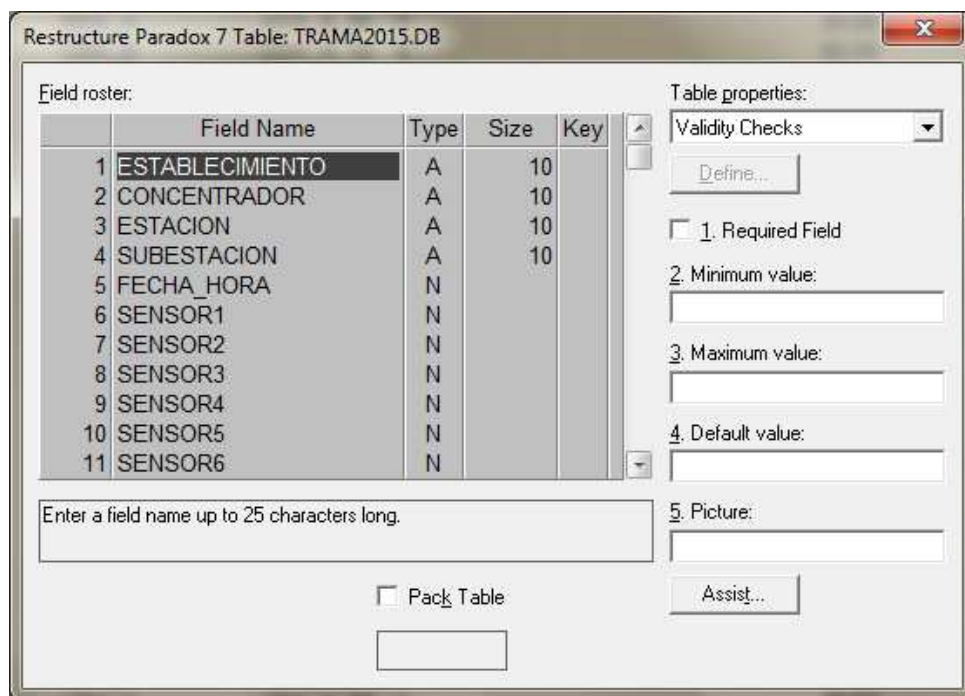
#### 5.4.3.2- Paradox

##### 5.4.3.2.1- Modelo de base de datos

Se mantiene el modelo de base de datos previamente planteado.

##### 5.4.3.2.2- Pasos realizados

La figura N°5.54 nos presenta en modelo de datos creado utilizando Paradox, en particular consta de una única tabla donde se almacenan los datos interpretados desde campo, tal como se comento en el apartado anterior.



**Imagen N°5.54 – Creacion de base de datos Paradox**

A continuación se detallan los campos de dicha base de datos:

- ESTABLECIMIENTO: Nombre de establecimiento, hasta 10 caracteres.
- CONCENTRADOR: Nombre de concentrador, hasta 10 caracteres.
- ESTACION: Nombre de estación, hasta 10 caracteres.
- SUBESTACION: Nombre de subestación, hasta 10 caracteres.
- FECHA HORA: Fecha y hora de los datos, proveniente de campo

- SENSOR 1 a 10: Valores registrados por los sensores.

Se lista a continuación el código para realizar la transacción en la base de datos, se puede observar que es muy similar al previamente implementado en Oracle

```

/*-----
insert sobre la tabla TRAMA2015 la cual registra toda la información que se recibe
de los sensores. Los datos se cargaran en TLabels y esos valores se insertan sobre
las tablas
----- */

Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("insert into TRAMA2015.db");
Query1->SQL->Add("(ESTABLECIMIENTO,      CONCENTRADOR,      ESTACION,
SUBESTACION, FECHA_HORA, SENSOR1, SENSOR2, SENSOR3, SENSOR4,
SENSOR5, SENSOR6, SENSOR7, SENSOR8, SENSOR9, SENSOR10)");
Query1->SQL->Add("values('"+ESTABLECIMIENTO-
>Caption+"', '"+CONCENTRADOR->Caption
      +", '"+      ESTACION-
>Caption+"', '"+SUBESTACION->Caption+"', '"+FECHAHORA->Caption+"', '"
+SENSOR1->Caption+"', '"+SENSOR2->Caption+"', '"+SENSOR3-
>Caption+"', '"+SENSOR4->Caption
      +", '"+SENSOR5->Caption+"', '"+SENSOR6-
>Caption+"', '"+SENSOR7->Caption+'", '"+SENSOR8->
Caption + ", '"+SENSOR9-
>Caption+"', '"+SENSOR10->Caption+'")");
Query1->ExecSQL();

Query1->Close();    //Propiedad Active=false
Query1->SQL->Clear();
Query1->SQL->Add("select * from TRAMA2015.db");
Query1->Open();    //Propiedad Active=true
}

```

#### 5.4.3.3- Nuevo modelo de datos.

La figura N°5.55 nos presenta en modelo de datos, es decir las tablas que se implementaran en el sistema.

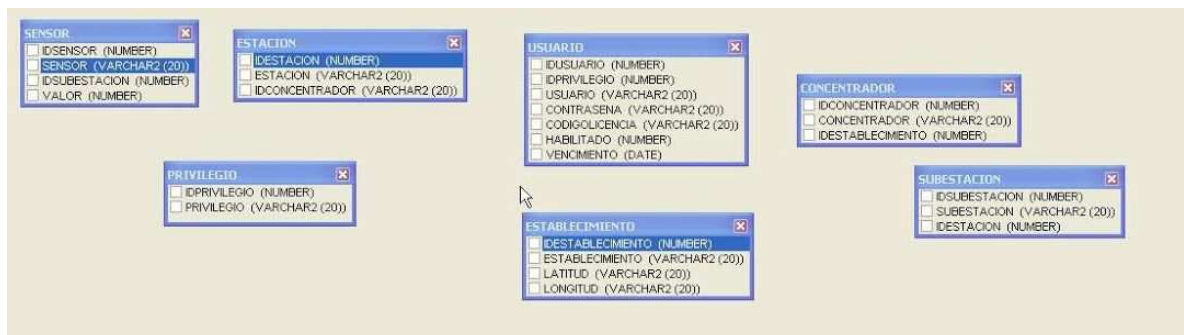


Figura N° 5.55 -Modelo de datos

#### **5.4.3.3.1- Descripción de tablas.**

##### **5.4.3.3.1.1- Tabla USUARIO.**

###### **Objetivo:**

Se registran los datos confidenciales del usuario. Tiene relación con la tabla Privilegio a través del campo IDPrivilegio.

###### **Campos:**

- IDUSUARIO: Es un ID identificatorio para el ingreso del usuario en sistema
- IDPRIVILEGIO: Identifica al tipo de usuario dentro del sistema
- USUARIO: Nombre de Usuario
- CONTRASENA: Contraseña del usuario
- CODIGOLICENCIA: Es el código de licencia que posee el usuario, el cual lo utiliza en su primer registro.
- HABILITADO: Si el usuario está habilitado o No:
- VENCIMIENTO: Es el vencimiento de la licencia, Una vez superada la fecha el usuario está inhabilitado.

##### **5.4.3.3.1.2- Tabla ESTABLECIMIENTO**

###### **Objetivo:**

Almacenamiento de los datos correspondiente a los establecimientos

###### **Campos:**

- IDESTABLECIMIENTO: Es una identificación única del establecimiento, que se utiliza en todo el sistema
- ESTABLECIMIENTO: Nombre de establecimiento.
- LATITUD: Identifica ubicación del establecimiento.
- LONGITUD: Identifica ubicación del establecimiento

##### **5.4.3.3.1.3- Tabla CONCENTRADOR**

###### **Objetivo:**

Almacén de datos de cada concentrador.

###### **Campos:**

- IDCONCENTRADOR: Es una identificación única del concentrador
- CONCENTRADOR: Nombre de concentrador
- IDESTABLECIMIENTO: Relaciona el establecimiento al que corresponde el concentrador

#### **5.4.3.3.1.4- Tabla ESTACION**

**Objetivo:**

Almacén de datos de cada estación

**Campos:**

- IDESTACION: Identificación única de la estación
- ESTACION: Nombre de estación
- IDCONCENTRADOR: Relaciona el concentrador al que corresponde la estación

#### **5.4.3.3.1.5- Tabla SUBESTACION**

**Objetivo:**

Almacén de datos de cada estación

**Campos:**

- IDSUBESTACION: Identificación única de la subestación
- SUBESTACION: Nombre de subestación
- IDESTACION: Relaciona este campo con la tabla ESTACION

#### **5.4.3.3.1.6- Tabla PRIVILEGIO**

**Objetivo:**

Almacén de datos de cada privilegio

**Campos:**

- IDPRIVILEGIO: Identificación única de privilegio. Tiene relación con la tabla USUARIO
- PRIVILEGIO: Nombre del privilegio dentro del sistema

#### **5.4.3.3.1.7- Tabla SENSOR**

**Objetivo:**

Almacén de datos de cada sensor.

**Campos:**

- IDSENSOR: Identificación única del sensor
- SENSOR: Nombre del Sensor.
- IDSUBESTACION: La pertenencia de este sensor con la subestación se señala en este campo.
- VALOR: Valor que devuelve el sensor. Útil para la visualización de los reportes.

## 5.5- Simulador para pruebas y desarrollo <C160sim.exe>

Con el fin de poder probar y estimular con información los software que se fueron desarrollando en paralelo al hardware del proyecto, se implementó un programa simulador que envía por los diferentes modos de comunicación tramas siguiendo el protocolo previamente visto en este mismo documento.

Se trata de un software programado en Borland C++ Builder por los motivos previamente comentados. Debe poderse conectar con el servidor de datos (ver apartado anterior) remotamente, mediante TCP/IP y el protocolo.

También presentará posibilidad de conexión por directa a al software Cliente final por puerto serie RS232, siguiendo también el protocolo de comunicación.

Por último tiene posibilidad de configurar las simulaciones de manera de tener un control de los estímulos.

A continuación se muestra la pantalla del simulador:

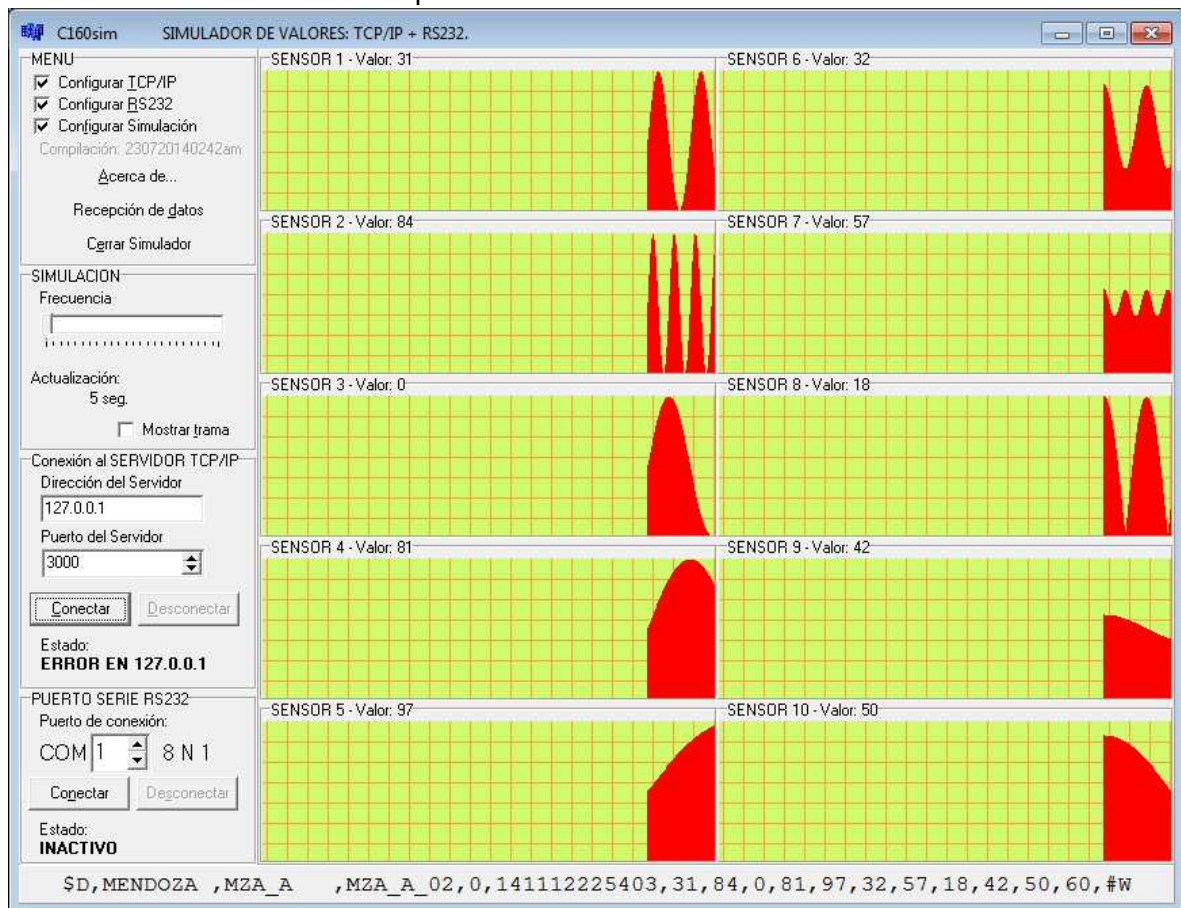


Imagen N°5.14 - Pantalla simulador de valores.



## **6- Pruebas del Sistemas**

Como a mediados del 2014 se tenía en funcionamiento, al menos parcialmente, las microestaciones, la estación colectora y el software, se puso como objetivo presentar el sistema en la ExpoProyecto 2014, exposición que tiene por objetivo dar a conocer las actividades de investigación y desarrollo realizadas en las distintas asignaturas de las carreras así como también las tareas que llevan a cabo los grupos de investigación del Departamento de Ingeniería e Investigaciones Tecnológicas. La misma se realizó a fines de Octubre del 2014.

Como objetivo para el resto del año y el 2015 se planteó finalizar con el sistema e implementarlo en sistemas de generación de energía renovable.

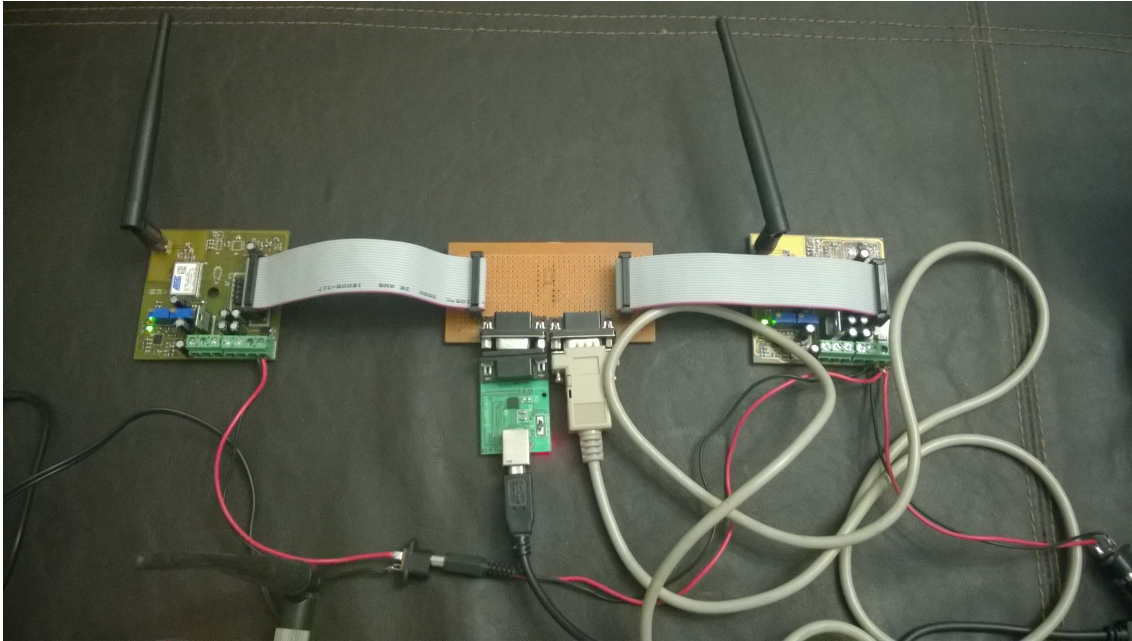
### **6.1- Descripción de la Demo**

A través del software de simulación, se generan una trama de datos según se describe en la sección 5.3.3.1, la cual es ingresada a una microestación a través del puerto serie, previamente adaptados los niveles de la PC a la microestación.

La microestación estará ejecutando un programa el cual tomara los datos del puerto serie y los retransmitirá. Los datos transmitidos serán recibidos por otra microestación, la cual los enviara a través de su puerto serie a la estación colectora.

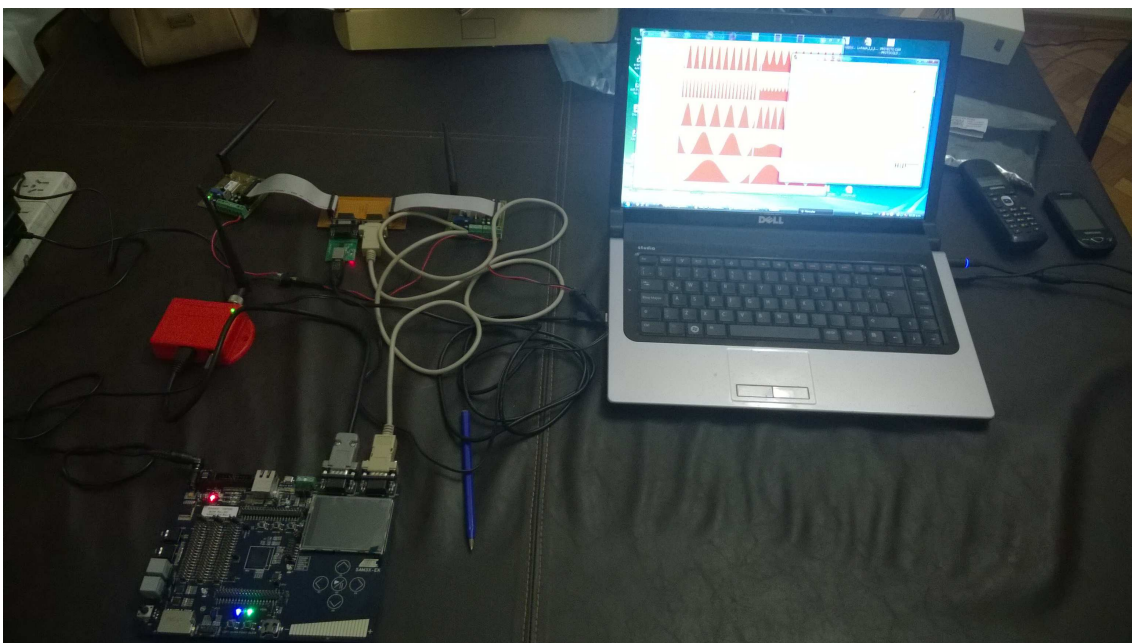
La estación colectora estar ejecutando un programa que analiza las tramas de datos recibidos y es un trama valida la transmitirá a través de un modem GSM/GPRS a un dirección IP y puerto previamente configurados. Donde se encontra un server el cual esta ejecutando software interprete y de base de datos para almacenar la información recibida y paralelamente estar atendiendo, por otro puerto, los requerimientos del software cliente.

La Figura N° 6.1 nos muestra las microestaciones conectadas a una placa adaptadora de niveles, la microestación de la izquierda estará recibiendo la trama de datos del simulador. Mientras que la microestación de la derecha recibirá los datos y los enviara por el puerto seria a la estación colectora.

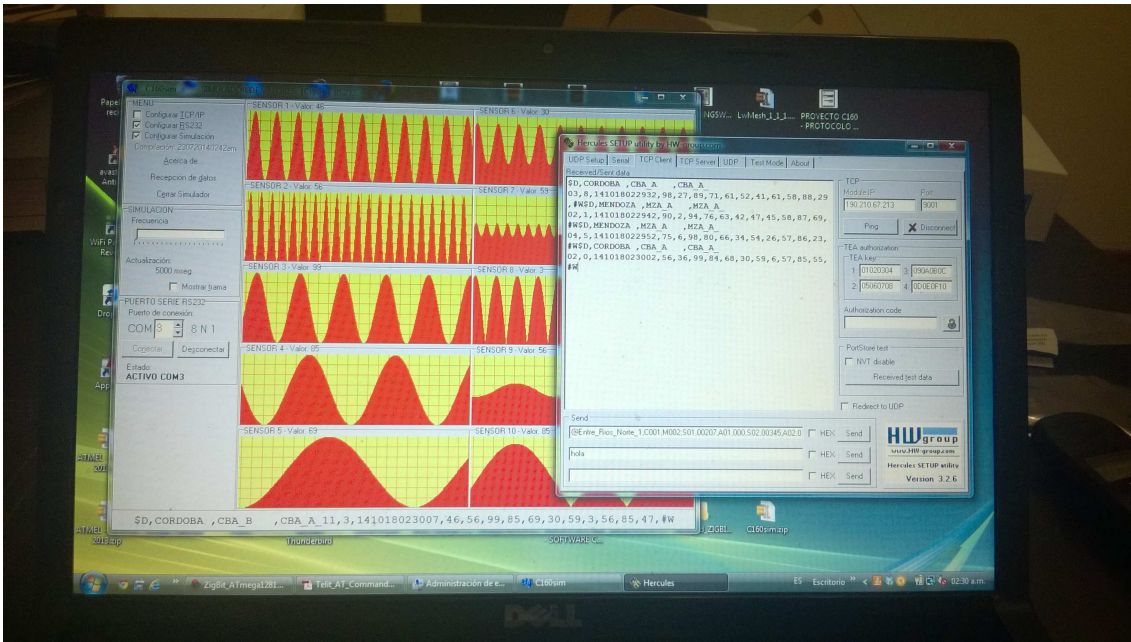


**Figura Nº6.1- Microestaciones**

La figura Nº 6.2 nos muestra el sistema completa, en donde se puede apreciar la estación colectora y el modem GSM/GPRS, así como también una PC que encuentra ejecutando el software de simulación y una aplicación cliente TCP, la cual se está conectando al servidor. El cual, para estas pruebas solo está haciendo un eco de los paquetes recibidos. En la figura 6.3 se puede ver esto último claramente.

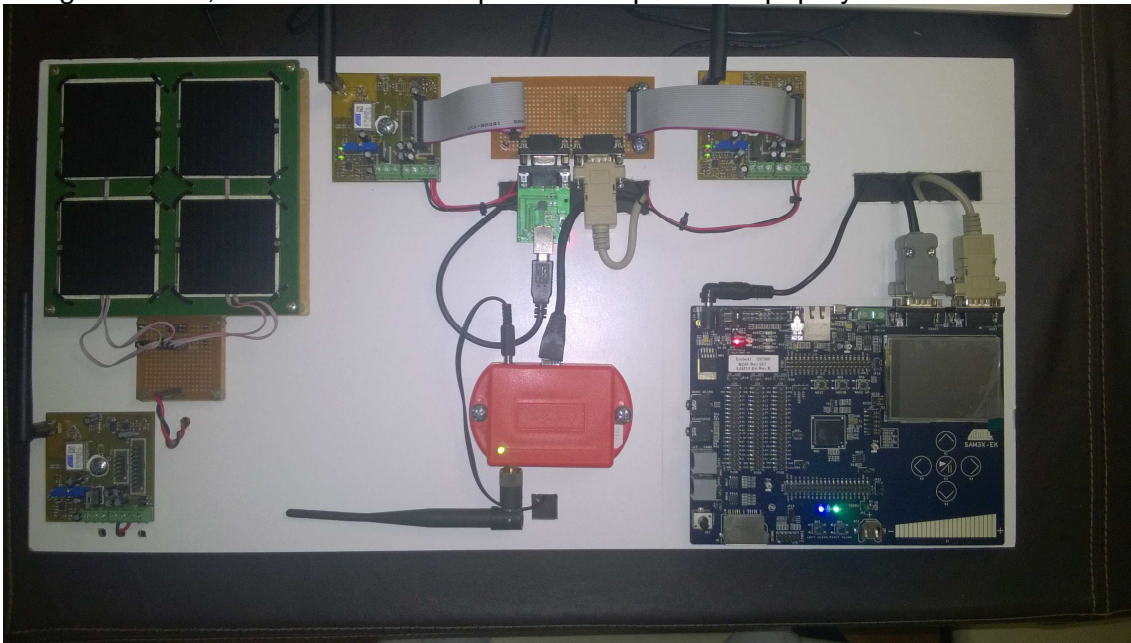


**Figura Nº 6.2- Sistema**



**Figura N°6.3. Simulador y Server**

La figura N° 6.4, muestra el sistema presentado para la Expoproyecto.



**Figura 6.4. Sistema montado para Expoproyecto.**

Las figuras N° 6.5, 6.6 y 6.7 muestran al sistema funcionando durante la Expoproyecto 2014.



**Figura Nº 6.4. ExpoProyecto 2014-1**



**Figura Nº 6.5 - ExpoProyecto 2014-2**



**Figura Nº 6.6 - ExpoProyecto 2014-3**

## **6.2- Sistemas de generación de energía renovable**

El proyecto planteo también como objetivos la medición de parámetros eléctricos, así como el estudio de energías renovables. El primero de ellos no llegó a concretarse del todo, mientras que en lo asociado al estudio de las energías renovables se cumplieron ampliamente los objetivos. Se desarrolló un sistema fotovoltaico con seguimiento de dos ejes, un sistema de biodigestión con control de temperatura y un aerogenerador. Implementándose en los dos primeros la telemetría de los magnitudes asociadas al proceso, sin llegar a implementarse todavía en el aerogenerador.

Dicha telemetría se pudo efectuar a raíz que los controles del sistema fotovoltaico así como del sistema de biodigestión son microcontrolados, pudiéndose conectar a una interfaz serial de estos los prototipos de las microestaciones.

En las secciones siguientes se presentan los distintos sistemas de energía renovables implementados.

## **7- Medición de Energía eléctrica.**

Un punto importante para obtener una mejora en aspectos energéticos es la medición de los consumos, lo cual junto con un análisis de la zona donde este emplazado el establecimiento permitirá una planificación de adecuada. Por lo tanto se pretende la obtención, ó lectura, de datos de los medidores de consumo ya existentes en la industria, así como su posterior envío a la estación colectora de datos, previamente procesados.

### **7.1- Parámetros de medición**

En la medición de consumo eléctrico se basa principalmente en la medición de la Potencia Activa (**P**) consumida. Algunos medidores también miden la Potencia Reactiva (**R**), y la Potencia Aparente (**A**). Las entidades de manufactura de electricidad cobran por la Potencia Activa, pero se mantiene controlada la reactiva por medio de la regularización del "coseno fi". Dicho parámetro debe ser tenido en cuenta en toda planta donde se utilicen gran número de motores eléctricos, ó generadores, ya que cualquier bobinado conectado a la línea desbalancea la línea.

### **7.2- Tipos de medidores**

Los medidores de consumo eléctrico existentes en Argentina poseen un amplio rango, ya que hay sitios donde todavía existen medidores electromecánicos, por ejemplo el medidor E111 de la firma ABB-Elster, que se publica en la página web de Edenor [31]. Para poder realizar una breve clasificación de los medidores según su propósito, se los puede agrupar de la siguiente forma:

- Medidores monofásicos o trifásicos, según el sistema de la red a través del cual obtienen la energía.
- Medidores de energía activa, reactiva ó aparente, según el tipo de energía que es capaz de medir
- Medidores de tarifa múltiple y de demanda máxima, según el horario de utilización y la carga máxima de corta duración.

Para este trabajo, la tercera clasificación no es de nuestro interés.

Otra posible clasificación es según su precisión. En la Argentina, dicho parámetro está regulado para los medidores de consumo eléctrico, debiendo ser Clase 1 o 2 (A o B). Para poder comparar los medidores en forma más eficiente, se presenta la Tabla 7.1, donde se agregó el tipo de salida que posee cada instrumento, por su importancia para el análisis de esta investigación.

Teniendo en cuenta que nuestro interés radica en el análisis de los datos extraídos de estos medidores, podemos observar que en todos los casos se encuentra un led indicador, el cual parpadea en forma directamente proporcional al consumo medido. Dicho led indica 1000 impulsos por Kilo Watt hora (kWh) consumido, excepto en el medidor ED110, cuya salida típica es de 10000 impulsos por kWh. La otra salida común en los medidores es por puerto óptico según IEC62056-21, ó por puerto IrDA. Los fabricantes suelen dar la opción de elegir entre una y otra norma.

A los efectos de esta investigación, no es relevante la posibilidad que dan algunos fabricantes de dispositivos para leer el consumo, y software para monitoreo de

consumos y confección de facturación, ya que se pretende monitorear estos equipos e incluir dichos datos en el software presentado en la sección 5.

Otro factor a tener en cuenta es que desde el Departamento de Energía y Ciencia, Tecnología e Innovación del Gobierno Nacional, se dio a conocer en Enero de 2013 dos proyectos para implementar Redes Inteligentes, como la clave para el uso racional de energía [32]. Con este objeto, las compañías se encuentran instalando medidores electrónicos. A ese momento, ya había alrededor de cien puntos de medición, distribuidos en provincias como Buenos Aires, Córdoba, San Juan y Tucumán.

Medidor	Marca	Tipo	Salida	Medición
<b>AMPy 5235</b>	AMPY	Monofasico	Led indicador 1000 imp/kWh	E. Activa
			Led indicador 1000 imp/kvarh	E. Reactiva
			Puerto Óptico según IEC62056-21, infrarrojo bidireccional	Máxima Demanda
<b>ACE3000</b>	ACTARIS	Trifasico	Led indicador 1000imp/kWh (IEC 62053-31)	Energia activa en 2 direcciones (P+, P-)
			Puerto Óptico según IEC62056-21 (former IEC 61107)	
<b>MS320</b>	SMC	Monofasico	Led indicador 1000 imp/kWh	E. Activa
			Puerto óptico IrDA (0,8 m)	
<b>HXE34</b>	Hexing-TSI	Trifasico	Led indicador 1000 imp/kWh	E. Activa
			Led indicador 1000 imp/kvarh	E. Reactiva
			Puerto Óptico según IEC62056-21, infrarrojo bidireccional	Valores Instantáneos / Máxima Demanda
<b>HEX12</b>	Hexing-TSI	Monofasico	Led indicador 1000 imp/kWh	E. Activa
			Led indicador 1000 imp/kvarh	E. Reactiva
			Puerto Óptico según IEC62056-21, infrarrojo bidireccional	Máxima Demanda
<b>ED110</b>	ZPA Smart Energy	Monofasico	Led indicador 10.000imp/kWh (configurable)	E. Activa
			Salida de pulsos SO según EN 62053-31	Valores Instantáneos y Máximos
			Puerto Óptico según IEC62056-21 (former IEC 61107)	Coseno Fi
<b>A100</b>	ABB Elster	- Monofasico	Led indicador 1000 imp/kWh	E. Activa
			IrDA TX	
<b>A102C</b>	ABB Elster	- Monofasico	Led indicador 1000 imp/KWh	E. Activa
			Salida de pulsos SO según IEC 62053-31	E. Reactiva
			IrDA TX (unidireccional)	
<b>A200</b>	ABB Elster	- Monofasico	Led indicador 1000 imp/kWh p/LCD	E. Activa
			Led indicador 1600 imp/kWh p/reg.mecanico	

**Tabla N°7.1-Medidores**

### 7.3- Planteo inicial

Se plantea efectuar la medición del consumo eléctrico a través de las siguientes interfaces:

- Led indicador de impulsos por kWh.
- Puerto óptico según IEC62056-21.

### 7.4- Medición a través del Led indicador

Debido a la dificultad y los costos que presentaba la implementación de un tablero, en el cual se pudiera conectar un medidor de energía eléctrica, las protecciones adecuadas y generar consumo eléctrico, se optó por simular los pulsos.

A fin de simular las condiciones de un medidor eléctrico comercial se reemplazó en un medidor estándar, fuera de servicio, el led de mismo por uno que pudiera ser controlado en forma externa, la figura N°7.1 y 7.2 muestra dicha implementación.



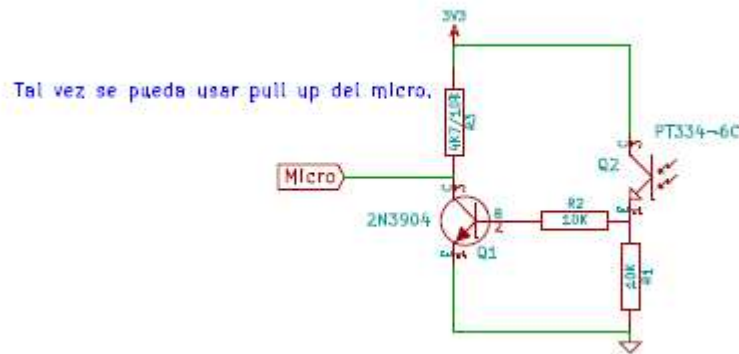
Figura N°7.1- Implementación de led indicador.



Figura N°7.2- Implementación de led indicador.



Para poder tomar los pulsos generados se implemento un circuito basado en un fototransistor PT334-6C [81], la figura N°7.3 muestra el circuito implementado.



. Figura N°7.3- Circuito de lectura de pulsos.

Por cuestiones personales del investigador involucrado no se pudo avanzar más de lo expuesto.

#### 7.5- Puerto óptico según IEC62056-21

Se comenzó con el estudio de la norma IEC62056-21:2002, la cual si bien no es la última vigente, pero se puede encontrar en internet, nos brinda una primera aproximación.

Mientras se estaba realizando dicho estudio se tuvo la oportunidad de conversar con los Ingenieros Roberto Clas y Jorge Salles de la empresa Elster [82], quienes nos comentaron que la comunicación de los equipos no era exactamente como definía la norma IEC62056-21 por lo menos a nivel protocolo, el cual era confidencial.

A su vez se estuvo en conversación con Ingenieros del departamento de investigación y desarrollo de la empresa Myeel [83], quienes tienen el desarrollo de una sonda de lectura para algunos modelos de la empresa Elster [84], nos comentaron que el desarrollo de dicho producto le llevo en cerca de 2 años.

En función de la información relevada se decidió dejar de lado la optencion de datos a través de dicha interfaz, principalmente porque no se contaba con el tiempo suficiente para relevar el protocolo de comunicaciones, y como segundo motivo esto no implicaría la compatibilidad con medidores de otros proveedores.

#### 7.6- Conclusiones

No se pudo cumplir con los objetivos planteados, las causas se deben a cuestiones personales del investigador, asi como también dificultades que no se habían contemplado al momento de la planificación del proyecto.

## **8.-Prototipo de luminaria publica con seguidor solar**

### **8.1- Introducción**

La iluminación es una de las formas más atractivas y fáciles de ahorrar dinero en la factura eléctrica. Al consumir casi el 20% de la electricidad total mundial, el hecho de substituir los sistemas de iluminación ineficientes u obsoletos por otras opciones ecológicas, tiene un impacto inmediato en el uso de la energía, las emisiones de CO<sub>2</sub>, y el medio ambiente. Casi dos tercios del alumbrado instalado en todo el mundo utilizan tecnologías anticuadas e ineficientes. Si se sustituyen estas fuentes de luz por luminarias LED, se puede llegar a obtener un ahorro energético promedio entre un 30% y un 40% [50]. Este es el objetivo que se persigue al utilizar luminarias Led de alta intensidad, las cuales, correctamente instaladas, pueden llegar a más de 50 mil horas de vida conservando más del 70% del flujo lumínico original, además de tener una elevada eficiencia lumínica si las comparamos con lámparas de vapor de sodio utilizadas en la actualidad.

La implementación del proyecto con un panel solar reside en las ventajas que posee este tipo de energía con respecto a otros tipos de generación de energía, tanto en lo económico, que si bien tiene un alto costo inicial después requiere un menor mantenimiento, como en otros aspectos: utiliza un tipo de energía inagotable, se puede usar en ubicaciones remotas, proporciona menos contaminación y ruido al ambiente. La energía del sol se recoge en paneles solares y se convierte en electricidad basándose en la aplicación del efecto fotovoltaico, que se produce al incidir la luz sobre unos materiales semiconductores, lo que genera un flujo de electrones en el interior del material, y en condiciones adecuadas, una diferencia de potencial que puede ser aprovechado con múltiples aplicaciones. Los paneles solares son inofensivos al medio ambiente, no generan polución y entregan energía limpia. Además, la energía solar ayuda a disminuir los gastos y a incrementar la independencia de los suministros tradicionales.

Es decir, se trata de una fuente de energía limpia, ecológica y económica, ya que después de la inversión inicial que supone instalar el equipo, sólo hay que aprovechar la energía que suministra el sol de forma natural y que en nuestro país es tan abundante, puesto que el mantenimiento de estos equipos consiste simplemente en la limpieza de los paneles. Lo que ocurre con este tipo de energía es que, siendo una alternativa viable, todavía no permite el uso indiscriminado de la electricidad. Así, cuando se instala este tipo de paneles, se aconseja usar esta energía fundamentalmente para la iluminación y tener una batería adicional para alimentar los aparatos eléctricos, sobre todo los de más potencia.

La cantidad de horas útiles de sol que pueden ser aprovechadas por el sistema dependen de la zona geográfica que se analice. El promedio de horas útiles anual en Buenos Aires es de casi 7 horas [51], por lo que es muy importante el agregado de un sistema de seguimiento solar con el fin de aumentar la eficiencia del sistema. Se verá mas adelante que con un sistema de seguimiento solar de 2 ejes como el que se utilizará en este proyecto, la captación solar aumenta un 30%.

### **8.2 - Descripción del sistema**

El sistema consta de un panel solar, el cual recibe radiación del sol durante el día y la almacena en la batería para su posterior uso durante la noche. A su vez, se utiliza un regulador con el objetivo de que la carga de la batería se haga a máxima transferencia de carga y se pueda controlar tanto la sobrecarga como la descarga excesiva de la batería, protegiendo así el sistema de acumulación de efectos perjudiciales para su

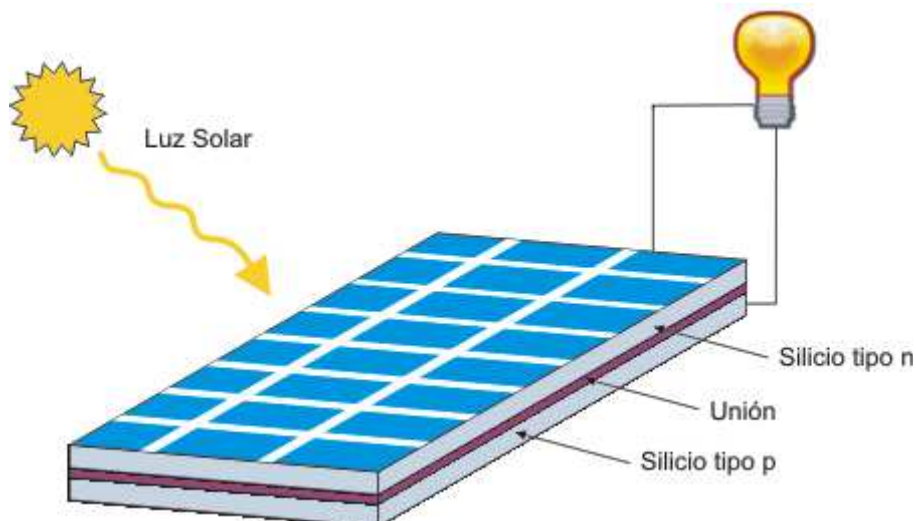
vida y funcionamiento. Con la ayuda de dicha batería se alimentarán dos módulos de LEDs de alta intensidad de 16 watts cada uno, que equivalen aproximadamente a una lámpara de vapor de sodio de 80 watts.

Para mejorar la eficiencia, el panel consta de un sistema de seguimiento solar, con el objetivo de obtener una mayor radiación y de esta manera una carga más rápida de la batería.

El diseño incluye el cálculo del panel solar, así como también el de la batería y la selección del regulador. Los módulos de leds empleados fueron desarrollados previamente como un trabajo independiente [52].

### 8.2.1- Paneles solares

Un panel solar es un módulo que aprovecha la energía de la radiación solar. Normalmente se refiere a los dispositivos para producir agua caliente (usualmente doméstica) y a los paneles fotovoltaicos utilizados para generar electricidad (Figura N°8.1).

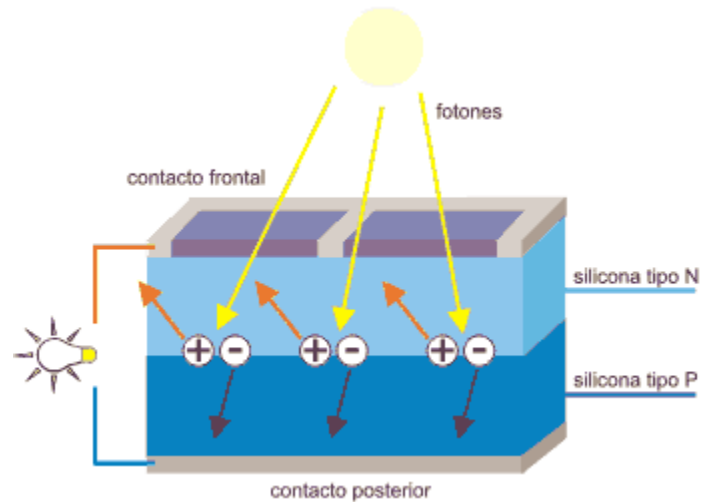


**Figura N°8.1-Panel Fotovoltaico.**

Los paneles fotovoltaicos están formados por numerosas celdas llamadas celdas fotovoltaicas, cuya función es convertir la energía de la luz en energía eléctrica, mediante un proceso denominado efecto fotovoltaico (FV). La luz solar está compuesta por fotones, o partículas energéticas. Estos fotones son de diferentes energías, correspondientes a las diferentes longitudes de onda del espectro solar. Cuando los fotones inciden sobre una celda FV, pueden ser reflejados, absorbidos, o pueden atravesarla. Únicamente los fotones absorbidos generan electricidad, esto es, la energía del fotón se transfiere a un electrón del material fotovoltaico. Con esta nueva energía, el electrón es capaz de escapar de su posición normal asociada con un átomo para formar parte de una corriente en un circuito eléctrico.

La parte más importante de las celdas solares, es el material fotovoltaico, el cual suele estar compuesto por capas de material semiconductor, en las cuales se produce la corriente de electrones. Estos semiconductores son especialmente tratados para formar dos capas diferentemente dopadas, tipo p y tipo n, como se observa en la Figura 8.2, para formar un campo eléctrico, positivo en una de las caras y negativo en la otra. Se puede ver que cuando la luz solar incide en la celda, se liberan electrones

que pueden ser atrapados por el campo eléctrico formando una corriente eléctrica. Debido al comportamiento de los semiconductores frente al aumento de la temperatura es que se utilizan para la construcción de los paneles solares.



**Figura N°8.2- Capas de semiconductor**

Además de las capas de semiconductores, las celdas solares están formadas por una malla metálica superior con el fin de recolectar los electrones de las placas de material semiconductor y transferirlos a la carga externa a través de un contacto para completar el circuito eléctrico. Los paneles solares son encapsulados con materiales transparentes, tales como vidrios, acrílicos, plásticos, etc., que permitan pasar la luz solar y proteger la integridad de los componentes internos del panel. Además se utiliza una capa antireflexiva para aumentar el número de fotones absorbidos.

### **8.2.2 - Seguimiento solar**

Un seguidor solar es un dispositivo mecánico capaz de orientar los paneles solares lo más perpendicular al sol posible a lo largo del día, lo que permite aumentar la eficiencia o rendimiento productivo a la pantalla fotovoltaica mediante el aumento de su área efectiva. Estudios han desvelado que un diseño mecánico optimizado, junto con un sistema electrónico de seguimiento robusto y altamente fiable, puede llegar a conseguir aumentos en la captación solar superiores al 30 % de media con respecto a sistemas fotovoltaicos sin seguimiento, con un incremento pequeño en el coste total de la instalación [53].

### **8.2.3 - Baterías (electricidad)**

Se denomina batería, batería eléctrica, acumulador eléctrico o simplemente acumulador, al dispositivo que almacena energía eléctrica, usando procedimientos electroquímicos y que posteriormente la devuelve casi en su totalidad; este ciclo puede repetirse por un determinado número de veces. Se trata de un generador eléctrico secundario; es decir, un generador que no puede funcionar sin que se le haya suministrado electricidad previamente mediante lo que se denomina proceso de carga.

#### **8.2.3.1 - Principios de funcionamiento**

El funcionamiento de un acumulador está basado esencialmente en un proceso reversible llamado reducción-oxidación (también conocida como redox), un proceso en

el cual uno de los componentes se oxida (pierde electrones) y el otro se reduce (gana electrones); es decir, un proceso cuyos componentes no resulten consumidos ni se pierdan, sino que meramente cambian su estado de oxidación, que a su vez puedan retornar al estado primero en las circunstancias adecuadas. Estas circunstancias son, en el caso de los acumuladores, el cierre del circuito externo, durante el proceso de descarga, y la aplicación de una corriente, igualmente externa, durante la carga. Resulta que procesos de este tipo son bastante comunes, por extraño que parezca, en las relaciones entre los elementos químicos y la electricidad durante el proceso denominado electrólisis, y en los generadores voltaicos o pilas.

Un acumulador es, así, un dispositivo en el que la polarización se lleva a sus límites alcanzables, y consta, en general, de dos electrodos, del mismo o de distinto material, sumergidos en un electrolito.

### **8.2.3.2 - Tipos de batería**

Existen varios tipos de baterías, a continuación se hace una breve descripción de este tipo de baterías:

#### **8.2.3.2.1- Batería AGM**

La gama AGM tiene una resistencia interna muy baja, por lo que son muy convenientes para usos que conllevan una alta intensidad de descarga, tales como inversores, propulsores y motores de arranque. Debido al uso de materiales de gran pureza y de rejillas de plomo-calcio, las baterías AGM tienen una autodescarga muy baja, lo que permite largos periodos de almacenamiento sin necesidad de carga.

#### **8.2.3.2.2- GEL**

La gama GEL ofrece la mejor durabilidad en ciclo profundo y la mayor vida útil. Las baterías de ciclo profundo guardan energía de modo que las fuentes de la corriente eléctrica las recarguen por medio de los alternadores, paneles solares, molinos de viento, etc. En sistemas de iluminación, las cargas aplicadas a las baterías guardan cierta relación con su capacidad y además suelen ser muy pequeñas al respecto de la capacidad de la batería. Se toma energía por debajo de la capacidad de la misma. Ejemplo, batería de 65 amperes, se toma 3 amperes a lo largo de 10 horas. Estas baterías son denominadas de ciclo profundo ya que admiten ser descargadas en un 90 % [5]. Debido al uso de materiales de gran pureza y de rejillas de plomo-calcio, las baterías GEL tienen una autodescarga muy baja, lo que permite largos periodos de almacenamiento sin necesidad de carga.

#### **8.2.3.2.3- MONOBLOCK**

Las baterías monoblock están destinadas a pequeñas instalaciones fotovoltaicas donde la relación calidad- precio tiene que ser equilibrada.

#### **8.2.3.2.4- OpzS**

Incluyen un pequeño orificio en su parte superior para realizar las medidas de tensión. Las baterías OpzS de FIAMM son ampliamente utilizadas en aplicaciones estacionarias. También se utilizan en aplicaciones cíclicas en las que se refieren cargas y descargas continuas.

En la figura N°8.3 podemos ver físicamente las baterías que hemos estado presentando



Figura N° 8.3 – Distintos tipos de baterías

#### 8.2.4 - LEDs

Se define por sus siglas como diodo emisor de luz, no es más que un pequeño chip de material semiconductor, que cuando es atravesado por una corriente eléctrica, en sentido apropiado, emite luz monocromática, es decir un componente electrónico semiconductor, con polaridad por lo que se usará en funciones de señalización, estética y, actualmente iluminación.

Los leds presentan muchas ventajas sobre las lámparas incandescentes o la iluminación con gas, tales como tubos y lámparas de emergencia. En la lista siguiente se enumeran algunas de estas ventajas.

- Larga vida útil
- Menor mantenimiento comparado con las fuentes de luz convencionales
- Alta eficiencia energética
- Extensa gama de colores por naturaleza
- Encendido instantáneo
- Gran robustez
- Luz directa
- Ecológico
- Ausencia de infrarrojos y ultravioletas

No se profundizara en ellas porque se puede encontrar gran número de bibliografía sobre estos [54].

#### 8.3 - Estudio de mercado

Al ser una tecnología relativamente nueva, más aún en nuestro país, el mercado está en una etapa de crecimiento. En la actualidad, existen cerca de una decena de empresas especializadas en la fabricación y/o comercialización de paneles solares en Argentina. Este es el caso de SunSolutions, que es una empresa dedicada a dar

soluciones en esta nueva forma de energía. Sin embargo, estas no cuentan con un sistema de seguimiento solar. Hay una empresa, La inesina solar, que si bien incorpora un seguidor solar, no se especializa en esta área, ya que posee soluciones para piscinas climatizadas, tecnología eólica, bombeo de agua solar, etc. En el exterior esta tecnología ha sido un poco más abordada, por lo que existen un mayor número de empresas candidatas a ser competidoras. Algunas de las que vamos a tener en cuenta en el estudio del mercado son SunTracking (España), Mecasolar (Estados Unidos, España, Grecia e India) y Solartiva (España) (Tabla N°8.1).

	SunTracking	Mecasolar	Solartiva
Ejes de seguimiento	2	2	2
Sup. Paneles (m2)	6	92 (máx.)	6
Dimensiones del seguidor (alto, ancho, profundo) (m)	3.5*2*0.7	6.7*12*7.7	3.10*2.30*0.5
Seguidores por motor	16	50	24
Máx. número de seguidores en serie	4	8	6
Ángulo de giro azimutal	180°	-120° a 120°	180°
Variación de ángulo cenital	50 °	60°	40°
Peso total (kg)	90	3000	80
Motor	Trifásico 380V	Trifásico 380V	Trifásico de 380V
Tipo de seguimiento	Astronómico por PLC	Astronómico por PLC	Astronómico por PLC
Sistema de protección por viento	si	si	si

**Tabla N°8.1 – Algunos productos en el mercado.**

El sistema de protección por viento consiste en un anemómetro el cual monitoriza continuamente la velocidad del viento de forma que para altas velocidades la estructura se recoge hasta una posición de mínimas cargas estructurales, con el fin de preservarla [55].

### 8.3.1- Necesidad del mercado

Los paneles solares son muy utilizados en lugares apartados de las grandes, medianas y pequeñas ciudades, donde no se cuenta con ningún tipo de energía. Estos sistemas suelen ser instalados para producir la energía necesaria para cubrir las necesidades básicas, como puede ser por ejemplo, la adquisición de agua de pozos por medio de una bomba eléctrica, la calefacción en invierno, la utilización de la energía para producir luz, etc. En lugares donde el sol no tiene tanta intensidad las baterías tardan bastante más en cargarse. Una probable solución para este problema es utilizar un seguidor solar para adquirir una mayor cantidad de energía.

Este proyecto apunta a la fabricación de un prototipo basado en las características de los dispositivos presentados, es decir, seguimiento solar mediante 2 ejes, con ángulos

de giros similares y, en principio, un control de posición de los ejes, con la idea de agregarle en un futuro un control de velocidad de viento, pero con una evidente reducción tanto en tamaño y peso, como en superficie de los paneles y número de seguidores. En esta etapa se apuntara a un mercado que utilice menor caudal de energía, por ejemplo para iluminación de perimetral, pero no por eso menos importante.

Se expone en la siguiente tabla N°8.2 las características del prototipo a desarrollado:

	Nuestro
Eje de seguimiento	2
Sup. Paneles (m2)	0.42
Dimensiones del seguidor (alto, ancho, profundo) (m)	1*0.9*0.9
Seguidores por motor	2
Máx. número de seguidores en serie	2
Ángulo de giro azimutal	180°
Variación de ángulo cenital	50°
Peso (kg)	50
Motor	MR8 VCC
Tipo de seguimiento	Por sensores
Control	de posición

**Tabla N°8.2 - Características del producto.**

#### **8.4 - Desarrollo**

Se observa en la figura N° 8.4, el diagrama en bloques del hardware electrónico que se desarrollará:

El bloque de control recibe información proveniente de los sensores de posición y los LDR, los cuales serán utilizados como sensores de intensidad, con esta información, el bloque de control intentara compensar en todo momento la posición para igualar la intensidad en ambos sensores, lo cual indicaría que el panel se encuentra perpendicular al sol. El movimiento para compensar la posición se hara por medio de dos motores de bajas revoluciones, mediante una etapa de potencia.

En paralelo con la etapa de control se encontrará funcionando la etapa de generación de energía e iluminación. En esta etapa el panel fotovoltaico capta la energía solar, la cual es enviada a la batería para su acumulación. Antes de ser enviada a la batería, la energía es regulada en tensión y corriente para una correcta carga de la batería. La luminaria se encenderá cuando un sensor, envíe la señal al detectar el nivel de luz correspondiente al encendido.



### 8.4.1 - Elección del panel solar

En primer lugar, se procede a definir dos términos importantes que, aunque suenan similares, son dos magnitudes diferentes.

**Irradiancia:** Es la magnitud que describe la radiación o intensidad de iluminación solar que llega hasta nosotros medida como una potencia instantánea por unidad de superficie,  $W/m^2$  o unidades equivalentes.

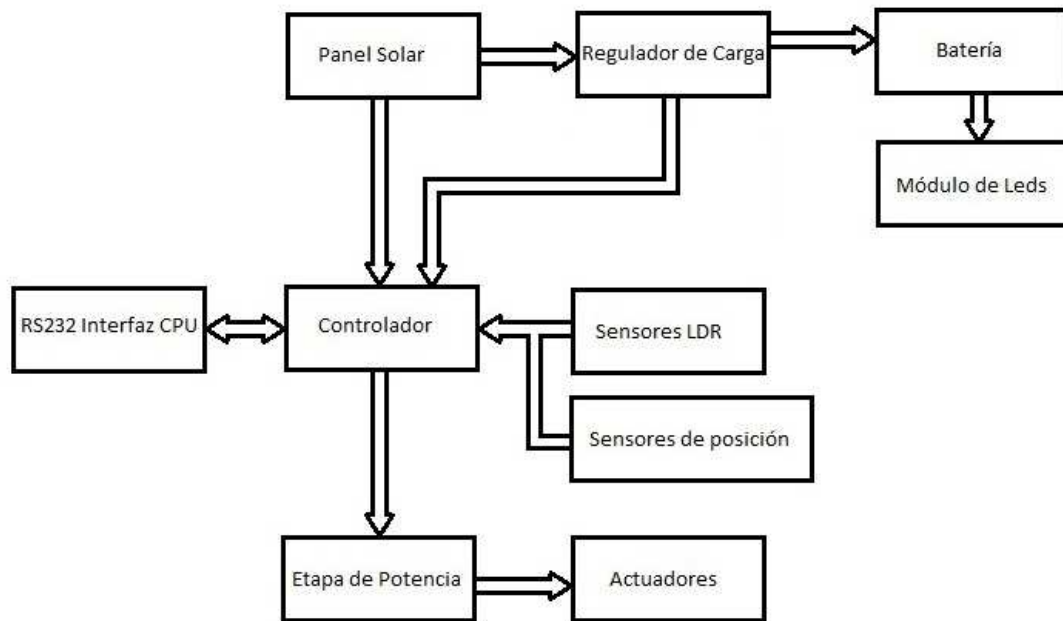


Figura N°8.4 – Diagrama en bloque.

**Irradiación:** Es la cantidad de irradiancia recibida en un lapso de tiempo determinado, es decir, la potencia recibida por unidad de tiempo y por unidad de superficie. Se suele medir en  $Wh/m^2$  o, en caso de un día, en  $Wh/m^2/día$  o unidades equivalentes.

Se debe introducir un concepto fundamental, las “Horas de Sol Pico” o HPS [horas]. Se puede definir el número de horas en que se dispone de una hipotética irradiancia solar constante de  $1000W/m^2$ . Es decir, una hora solar pico “HPS” equivale a  $1 KWh/m^2$  o, lo que es lo mismo,  $3,6 MJ/m^2$ . Este es un modo de contabilizar la energía recibida del sol agrupándola en paquetes, siendo cada paquete de 1 hora recibiendo  $1000 Watts/m^2$ .

Para calcular entonces el valor de HPS se debe dividir el valor de la irradiancia incidente entre el valor de la potencia de irradiancia en condiciones estándar de medida (STC), pues es en esas condiciones donde se cumplen las características eléctricas de los módulos fotovoltaicos. Ese valor de irradiancia en condiciones estándar de medida es de  $1000 Watts/m^2$ . Es decir, si se dispone de los datos de irradiancia solar de un determinado día y se divide entre  $1000$ , se obtienen las HPS. Por ejemplo, si se tiene una irradiancia de  $3800 Wh/m^2$ , para pasarla a HPS, se divide entre  $1000 W/m^2$ , con lo que se obtiene  $3,8 HPS$ .

Los pasos a seguir para dimensionar el sistema fotovoltaico autónomo son [56]:

- Estimación del consumo.

- Datos del lugar donde se realizará la instalación para saber la irradiación de la que disponemos.
- Dimensionado del generador fotovoltaico (número de paneles necesarios).
- Dimensionado del sistema de acumulación (número de baterías).
- Dimensionado del regulador.

Una vez definidos los pasos, exponemos el método de cálculo .

Unidades	Carga	Potencia Unitaria (Watt)	Hora de Funcionamiento al día (Horas)	Total de energía necesaria (Wh)	Total de energía necesaria (Wh) x Margen de Seguridad (10%)
2	Módulo de leds	16	10	320	352
			<b>TOTAL</b>	<b>320 Wh/día</b>	<b>352 Wh/día</b>

**Tabla N°8.3 - Consumo.**

#### 8.4.2 - Estimación del consumo

Con los datos de la Tabla N°8.3, "Tabla de Consumos" se obtiene el consumo medio diario de la instalación al que se ha aplicado un 10% como margen de seguridad recomendado.

Se debe también tener en cuenta que en la instalación habrá pérdidas por rendimiento de la batería y esto influye en la energía necesaria final.

Las baterías selladas que se producen actualmente, garantizan un rendimiento del 95%. Para el buen dimensionado, se tomará un rendimiento de los conductores de un 100%.

Así pues para el cálculo de los consumos medios diarios (Lmd) se considera la siguiente expresión (Formula N°8.1):

$$L_{md} = \frac{L_{md,DC}}{bat \times con} = \frac{352}{0.95 \times 1} = 370.5 \text{ Wh/día} \quad (8.1)$$

Siendo (Lmd) el consumo medio de energía diario y (Lmd,DC) el consumo de energía diario de las cargas en continua.

O bien, si se quiere expresar como el consumo de energía medio en Ah/día(Fórmula N°8.2)

$$Q_{Ah} = \frac{L_{md}}{V_{bat}} = \frac{370.5}{12} = 30.9 \text{ Ah/día} \quad (8.2)$$

Como era de esperarse, el consumo medio diario real es ligeramente superior al nominal, pues como se mencionó, se ha tomado en cuenta las pérdidas que se pueden producir en algunos elementos de la instalación y el margen de seguridad del 10%.

Como dato adicional, se podría calcular el consumo total anual (LT) y medio anual (Lma). En este caso coincide con el medio diario (Lmd), pues el consumo que se ha

estimado es constante todo el año, no sucedería así si hubiera variaciones de consumos estacionales (Formula N° 8.3 y 8.4)

$$LT = L_{md} \times 365 \text{ días} = 135232.5 \text{ Wh/Año} \quad (8.3)$$

$$L_{ma} = L_{md} = \frac{LT}{365 \text{ días}} = 370.5 \quad (8.4)$$

### 8.5 - Dimensionado del generador fotovoltaico.

Una vez obtenida la energía suministrada por cada panel se puede calcular la cantidad de paneles necesarios dividiendo la potencia necesaria diaria por la energía diaria generada por cada panel (Formula N°8.5)

$$N_T = \frac{L_{md}}{E_p} = \frac{L_{md}}{P_{MPP} \times HPS} = \frac{370.5}{60 \times 4.5} = 1.37 \rightarrow 2 \text{ Paneles Solares} \quad (8.5)$$

Al observar el resultado obtenido en la ecuación (5), notamos que utilizando 2 paneles al segundo panel lo estaríamos desaprovechando al utilizarlo sólo a un 37% de su capacidad. Como el sistema contará con un sistema de seguimiento solar, como se mencionó anteriormente, con la implementación de un seguidor solar se obtiene un aumento de eficiencia entre un 30% y un 40% [57][58]. Recalculando se obtiene(Fórmula N° 8.6):

$$N_T = \frac{L_{md}}{P_{MPP} \times HPS} = \frac{370.5}{60 \times 4.5 \times 1.4} = 0.98 \rightarrow 1 \text{ Panel Solar} \quad (8.6)$$

De esta manera, será posible utilizar un solo panel de 60W.

#### 8.5.1 - Dimensionado del sistema de acumulación.

Para el dimensionamiento de la batería los parámetros a considerar son la máxima profundidad de descarga (estacional y diaria) y el número de días de autonomía. Como norma general, se tomaran estos parámetros:

- Profundidad de Descarga Máxima Estacional (Pdmax,e) = 70% = 0.7
- Profundidad de Descarga Máxima Diaria (Pdmax,d) = 40% = 0.4
- Número de días de Autonomía (N) = 1

La profundidad de descarga es el valor, en porcentaje, extraída de un acumulador plenamente cargado en una descarga. Los reguladores limitan esta profundidad, y se calibran habitualmente para permitir profundidades de descarga de la batería en torno al 70%. Dependiendo de la máxima profundidad de descarga permitida, el número de ciclos de carga y descarga durante toda la vida útil de la batería será mayor o menor [59].

Las profundidades de descarga máximas se consideraran en el sistema para un ciclo diario (profundidad de descarga máxima diaria) estará en torno al 40%.

Se calcula la capacidad nominal necesaria de la batería en función de la profundidad de descarga estacional y diaria. La mayor de ellas será la que seleccione, pues de lo contrario se puede incurrir en una insuficiencia estacional o diaria (Formulas N° 8.7 y 8.8).

Capacidad nominal de la batería en función de la descarga máxima diaria (Cnd) :

$$C_{nd}(Wh) = \frac{L_{md}}{P_{Dmaxd} \times f_{ct}} = \frac{370.5}{0.4 \times 1} = 926.3 Wh \quad (8.7)$$

$$C_{nd}(Ah) = \frac{C_{nd}(Wh)}{V_{bat}} = \frac{926.3}{12} = 77.2 Ah \quad (8.8)$$

La explicación de las dos ecuaciones es sencilla, se necesita que la batería utilizada entregue una energía diaria Lmd donde se permite solamente un 40% de descarga máxima diaria y se supone un Factor de corrección de Temperatura unitario (Fct=1), es decir que se desprecia el efecto que la temperatura tiene sobre la batería. Una vez conocida la energía en Wh de la batería, simplemente se divide por la tensión de la misma (12V en este caso) y ya tenemos la capacidad mínima que necesitamos para nuestro sistema de acumulación en función de la descarga máxima diaria.

Capacidad nominal de la batería en función de la descarga máxima estacional (Cne) (Formulas N° 8.9 y 8.10):

$$C_{ne}(Wh) = \frac{L_{md} \times N}{P_{Dmaxe} \times f_{ct}} = \frac{370.5 \times 1}{0.7 \times 1} = 529.3 Wh \quad (8.9)$$

$$C_{nd}(Ah) = \frac{C_{ne}(Wh)}{V_{bat}} = \frac{529.3}{12} = 44.1 Ah \quad (8.10)$$

La explicación es similar a la anterior, necesitamos que la batería utilizada entregue una energía diaria Lmd donde podamos disponer de ella durante 1 día sin sol, sin permitir una descarga mayor del 70% y se supone un Factor de corrección de Temperatura unitario (Fct=1). Una vez sabida la energía en Wh de la batería, simplemente dividimos entre la tensión de la misma (12V en este caso) y ya tenemos la capacidad mínima que necesitamos para nuestro sistema de acumulación en función de los días de autonomía.

Así pues se escogera la mayor, es decir la capacidad nominal de las baterías sería, como mínimo, C100=77.2 Ah.

Por lo tanto, se escoge una batería de GEL de 12V x 90Ah ya que la misma presenta grandes ventajas entre las cuales se puede mencionar una autodescarga muy baja debido al uso de materiales de gran pureza y de rejillas de plomo-calcio, lo que permite largos periodos de almacenamiento sin necesidad de carga.

Las baterías de gel tienen mejor duración de vida en modo flotación. El electrolito se absorbe por capilaridad en una estera en fibra de vidrio situada entre las placas.

VRLA GEL: duración de 12 años

Características técnicas [60]:

- Peso (Kg): 26
- Dimensiones mm: 350x167x183
- Voltaje: 12V
- Capacidad de la batería: 90Ah C10

Observaciones: Garantiza el mejor contacto posible y eliminan la necesidad de bornes de batería. Fabricadas de acuerdo con la certificación ISO 9002, las baterías cumplen con las normativas CE y UL y sus recipientes son de ABS resistente al fuego.

### 8.5.2 - Dimensionado del regulador.

Para el dimensionado del regulador se debe calcular cuál será la máxima corriente que debe soportar el regulador a su entrada y a su salida. Para calcular la corriente de entrada al regulador se hace el producto de la corriente de cortocircuito del módulo fotovoltaico y un factor de seguridad. La corriente de corto circuito según la hoja de datos del fabricante del panel solar es de  $I_{sc}=3.64$  A y el factor de seguridad tomado es de un 25% por encima del valor (Fórmula N° 8.11).

$$I_{entrada} = 1.25 \times I_{MOD,sc} = 1.25 \times 3.64 = 4.55 \text{ A} \quad (8.11)$$

Siendo, ( $I_{MOD,SC}$ ) la corriente unitaria del módulo fotovoltaico en condiciones de cortocircuito. Se usa la corriente de cortocircuito para el cálculo de la corriente de entrada al regulador porque será la máxima corriente que podría ser generada por el módulo fotovoltaico, debe ser esa la que tengamos en cuenta para evitar pérdidas de rendimiento.

El factor de seguridad para evitar daños ocasionales al regulador es 1,25.

Para el cálculo de la corriente de salida se deben considerar las potencias de las cargas DC, es decir la potencia necesaria para el funcionamiento de los motores, la luminaria y la carga de la batería (Fórmula N° 8.12 y 8.13):

Potencia luminarias: 32W  
 Potencia motor y control: 8,78W  
 Potencia carga de 30atería: 12W

Entonces:

$$P_{DC} = 32 + 8,78 + 12 = 52,78\text{W} \quad (8.12)$$

$$I_{SALIDA} = (1.25 * P_{DC}) / V_{bat} = (1.25 * 52,78) / 12\text{V} = 5,5\text{A} \quad (8.13)$$

Siendo  $P_{DC}$  la potencia de las cargas en continua.

El regulador debería soportar una corriente, como mínimo de 4,55 A a su entrada y 5,5 A a su salida.

Para la selección del regulador se tuvo en cuenta las siguientes características de los paneles solares:

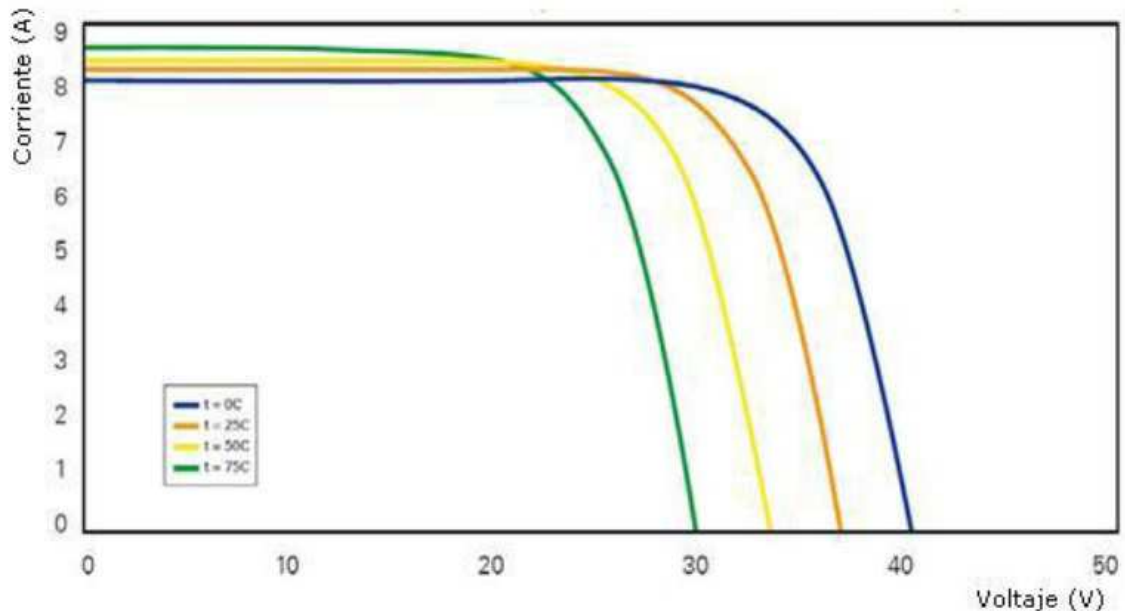
- La tensión de salida en circuito abierto ( $V_{oc}$ ) varía en función de la temperatura (a menor temperatura mayor es la tensión de salida).

- La corriente entregada por el panel es proporcional a la intensidad de radiación y a la superficie del panel.

Curvas de respuesta del panel solar: respuestas con la temperatura (Figura N° 8.5) y respuestas con la irradiación solar (Figura N° 8.6)

Esta característica de los paneles hace que el punto de máxima potencia que puede suministrar el panel se vaya desplazando y si no se lo puede seguir se tendrá una disminución del rendimiento del regulador, con lo cual tenemos que emplear la técnica del seguimiento del punto de máxima potencia que se explica a continuación.

El seguimiento del punto de máxima potencia, conocido normalmente como MPPT (Maximum Power Point Tracking) es un sistema electrónico que gestiona los módulos fotovoltaicos de tal manera que permite extraer la máxima potencia de éstos. El MPPT no es un seguidor solar, es un sistema totalmente electrónico que varía el punto eléctrico de operación de los paneles y así permite que el panel entregue la máxima potencia extraíble. La potencia adicional extraída se convierte en un incremento de la intensidad de carga de la batería.



**Figura N°8.5 - Respuesta con la temperatura.**

Para entender el funcionamiento de un MPPT, se considera primero el funcionamiento de un regulador de carga tradicional. Cuando un regulador de carga normal está cargando una batería descargada, lo que hace es simplemente cerrar un circuito que permite conectar los paneles con la batería. Esto, "fuerza" a los paneles a trabajar a la tensión de operación de la batería, lo que normalmente no coincide con la tensión de máxima potencia ideal del panel. La curva de Potencia/Tensión/Intensidad para un panel tipo de 75 W en condiciones de temperatura de 25°C y 1000W/m<sup>2</sup> de irradiancia, es la que se ve en el Figura N°8.7.

El regulador de carga tradicional conecta la batería con el módulo, y por tanto fuerza al módulo a trabajar a 12 V. Este hecho de hacer que el módulo de 75 W trabaje a 12 V, lo que provoca es que de forma artificial el módulo simplemente entregue unos 53 W.

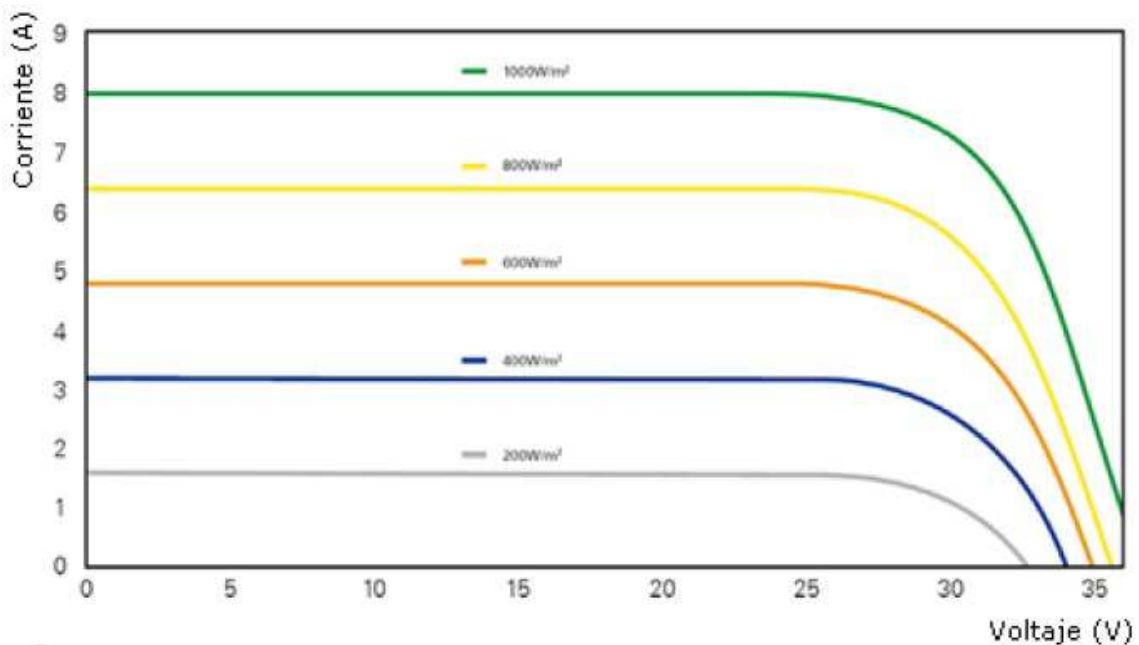
Los reguladores de carga que incorporan MPPT, lo que hacen es calcular en todo momento la tensión a la cual el módulo es posible que produzca la máxima potencia para las condiciones del momento. En el grafico anterior la tensión de máxima

potencia ( $V_{mp}$ ) es 17 V, y lo que el regulador de carga con MPPT hace es independizar la tensión de las baterías de la del módulo, dejando que el módulo fotovoltaico opere a 17 V independientemente de la tensión de la batería, consiguiendo un aumento de la intensidad de carga de la batería. Esto lo realiza, de forma genérica, con un convertidor DC/DC de alta eficiencia que convierte los 17 V a la entrada del regulador, en los aproximadamente 12 V a los que trabaja la batería en la salida de este. Suponiendo que no hubiese pérdidas y que la eficiencia fuese del 100% entre la salida del módulo y la entrada de la batería se tendría que la intensidad de carga de la batería sería:

La corriente de cortocircuito del panel solar tiene un valor de  $I_{sc}=3.64A$  Por lo tanto tendremos (Formula N°8.14):

$$V_{mp}/V_{bat} \times I_{mod} = 17V/12V \times 3,64A = 5,15A \quad (8.14)$$

Esto se traduce en un incremento de 1,51A lo que significa un 41,5% más, cifra nada despreciable [61].



**Figura N°8.6 - Respuesta con la irradiación solar.**

### 8.5.3 - Regulador elegido [62]

Se utilizó un regulador de 12V y 10 A de la empresa Electroimpulso (Figura N° 8.8) para la carga de las baterías. Este regulador es utilizado específicamente en instalaciones de energía solar. Tiene un rango de tensión de entrada de entre 14V y 28V, suministrando una tensión de salida regulable de entre 13V y 15,5V.

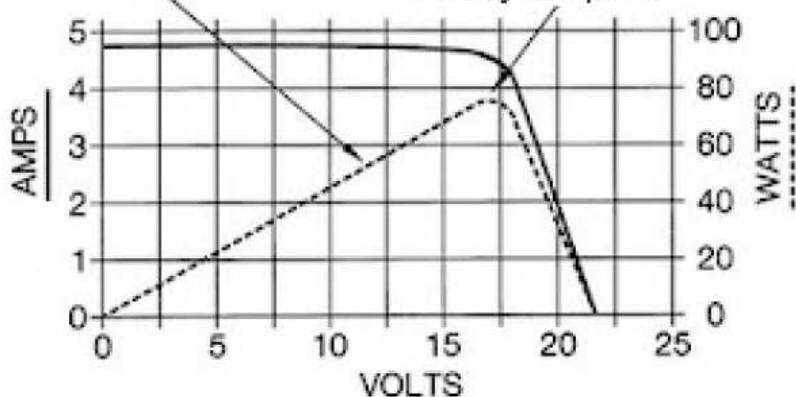
El cargador funciona en modo corriente constante hasta alcanzar el voltaje máximo de la batería, momento en el que se enciende el led verde y cambia a modo pwm de voltaje constante.

El led amarillo enciende cuando la batería tiene al menos 12V de tensión. Si se apaga el led amarillo significa que la tensión de la batería es menor a 10V. Es recomendable

desconectar los consumos del sistema con el fin de proteger la batería de una descarga demasiado profunda que pueda dañarla

Los cargadores convencionales cargan a 12V, solo extraen al rededor de 53W.

Este sistema hace trabajar al panel en su punto de máxima potencia, extrallendo los 75W que es capas de entregar el panel



curva tension/corriente/potencia de panel solar de 75W en condiciones normales de ensayo

**Figura Nº8.7- Curva de Potencia, Tensión e Intensidad.**



**Figura Nº8.8 - Regulador de tensión .**

Las especificaciones del mismo son las siguientes:

- Corriente máxima de funcionamiento: 10<sup>a</sup>.
- Tensión de entrada: 14V a 28V.
- Tensión regulada de carga de batería: 14V – dispone de preset para ajuste fino. (13 – 15,5V)
- 3 leds indicadores del estado del sistema.
- Módulo en aluminio anodizado, sellado con resina con frente plastificado.
- Protegido contra sobre corriente mediante fusibles
- Protegido contra inversión de voltaje de entrada mediante diodos antiparalelo y fusibles

El código de los leds indicadores de estado se presenta en el grafico Nº 8.9.

Su instalación es muy sencilla y se puede observar su conexión en la figura Nº 8.10.



## 8.6 - Sensores utilizados en el seguimiento solar

Para la implementación del seguimiento solar utilizaremos LDR (Light Dependent Resistor o resistor dependiente de la luz) como sensores de la captación solar, visualizado en la figura N° 8.11. Como su nombre lo indica, estos componentes varían su resistencia eléctrica en función a la luz recibida sobre su cara plana. Si bien los valores que puede tomar un LDR en total oscuridad y a plena luz puede variar un poco de un modelo a otro, en general oscilan entre unos 50 a 1000 ohms cuando están iluminadas (por ejemplo, con luz solar) y valores comprendidos entre 50 Kohms y varios Mohms cuando está a oscuras. Desde el punto de vista constructivo, los LDR están fabricados con materiales de estructura cristalina, siendo los más utilizados el sulfuro de cadmio y el seleniuro de cadmio, aprovechando sus propiedades fotoconductoras.

LEDS	LEDS ENCENDIDOS	ESTADO DEL SISTEMA
ROJO		Cargando, voltaje de batería por debajo de 12V
VERDE		
AMARILLO		
ROJO		Cargando, voltaje de batería mayor o igual a 12V
VERDE		
AMARILLO		
ROJO		V. máx. alcanzado, modo pwm tens. Constante
VERDE		
AMARILLO		

Figura N°8.9 – Código de leds indicadores de estado .

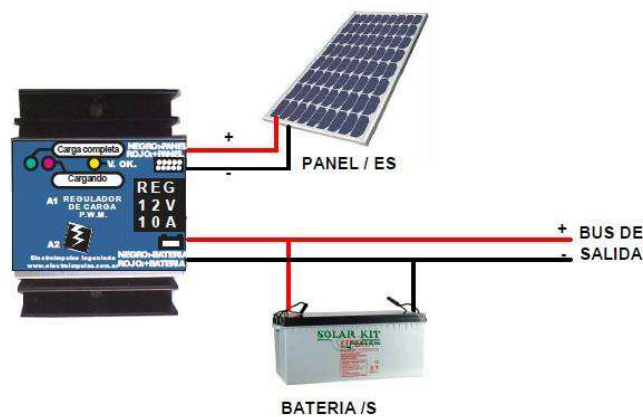


Figura N° 8.10 - Instalación del regulador



Figura 8.11 - LDR.

Para finalizar, el sistema de seguimiento solar posee dos encoders con el fin de poder determinar la posición de cada uno de los ejes en un instante determinado. Luego, mediante una comunicación RS-232 visualizaremos estos valores en una computadora.

### **8.7 - Elección del Microcontrolador**

Para la selección del microcontrolador a utilizar, primero nos plantearemos las necesidades que deberá cumplir:

- Permitir el trabajo con PWM.
- Sencilla programación y conexión a la PC.
- Permitir el trabajo con interrupciones y timers.
- Disponer de varios puertos de entradas/salidas – analógicas/digitales.
- Contar con un conversor analógico/digital incorporado.
- Precio adecuado al sistema implementado.

Luego de un estudio de mercado de los distintos microcontroladores existentes, decidimos utilizar el Microcontrolador PIC 16F877A de Microchip debido a que brinda correctas características de funcionamiento a un costo razonable. Su principal ventaja es que presenta un bajo consumo de corriente (< 0.6 mA típico @ 3V, 4 MHz – 20 uA típico @ 3V, 32 KHz - < 1 uA corriente típica en reposo), siendo de vital importancia para mejorar la eficiencia energética del sistema. Al ser un Microcontrolador de 40 pines nos ofrece 33 pines utilizables como entradas/salidas. Presenta una alta velocidad de operación (DC: 20MHz entrada de clock – 200 nS por ciclo de instrucción). Se programa en lenguaje C++. Su pinout es compatible con otros Microcontroladores de la misma familia, los cuales presentan la ventaja de brindar mayor capacidad de memoria y velocidad de operación, esto podría ser útil si en un futuro se introducen mejoras en el sistema y se requiere mayor capacidad de procesamiento. Permite trabajar con PWM, el cual es utilizado para comandar los motores paso a paso, ser reprogramado vía interface RS232 siendo una interfaz rápida y sencilla e independizándose de contar con un programador, y trabajar con interrupciones y timers. Posee 8Kx14 words de memoria programable FLASH, 368x8 bytes de memoria de datos (RAM) y 256x8 bytes de memoria de datos EEPROM.

Estas características son más que suficientes para la aplicación que vamos a implementar.

Para la implementación, utilizamos una placa de desarrollo multipropósito que contiene al Microcontrolador PIC mencionado anteriormente, con módulos de entradas/salidas digitales y analógicas, salida para Display lcd, servomotor, PWM y conexión RS232.

### **8.8 - Motores utilizados**

Para este proyecto es indispensable la utilización de motores de muy baja velocidad debido a la velocidad necesaria para que la estructura siga el trayecto del sol.

Los motores elegidos para producir el movimiento de los dos ejes, son los motorreductores de la empresa Ignis modelo MR83-1,2. Debido a sus características, que se amoldan a las necesidades del proyecto.

Estos motores se alimentan con una tensión nominal de 12V, presenta una corriente al vacío de 0,2 A una corriente nominal de 0,99 A y una corriente de arranque de 7,92 A. Su velocidad nominal es de 6140 RPM sin el reductor mientras que con el

reductor la velocidad baja a 1,2 RPM. Su cupla nominal es de 30 Kgf/Cm y su cupla de arranque es 120 Kgf/Cm. En la figura N° 8.12 se puede observar las medidas del motor.

Se resalta en naranja el modelo de motor utilizado en el proyecto, en verde se resalta el tipo de reducción que utiliza, donde la relación del desmultiplicador es de 5184:1 y en azul la velocidad nominal con el reductor.

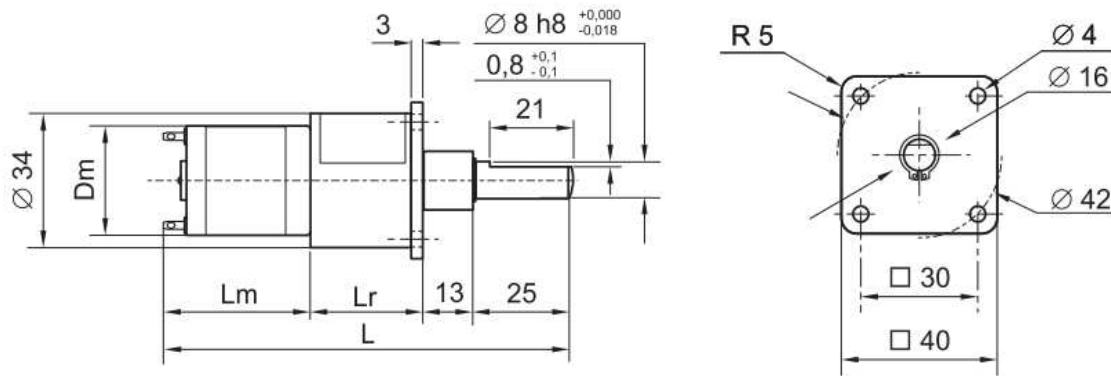


Figura N° 8.12 - Motor MR83-1,2.

En la tabla N° 8.4 se observa con detalle las características del motor elegido

				Modelo		
MR - Eje - VEL - Motor				MR83-VEL	MR86-VEL	MR85-VEL
Servicio				Normal		
Potencia [Hp]				0,0067	0,0107	0,0309
Tensión nominal [VCC]				12		
Io (Intensidad vacío). Inom (nominal). Is (arranq/bloq.) [A]				0,20 . 0,99 . 7,92	0,10 . 0,48 . 3,05	0,5 . 1,5 . 6,5
RPM Nom . RPM Vacío (motor)				6140 . 7600	7430 . 8600	7810 . 9100
Peso . Adicional por etapa [Kg]				0,160	0,005	0,270 . 0,010
Etapas	Desmult.	Engran.	VELocidad (RPM)	Cupla [Kgf.cm]		
0	1:1	0	6400	0,077	0,103	0,284
1	4:1	4	1600	0,31	0,41	1,13
1	6:1	6	1066	0,46	0,62	1,70
2	16:1	44	400	1,23	1,65	4,54
2	24:1	64	266	1,85	2,47	6,80
2	36:1	66	177	2,77	3,71	10,21
3	64:1	444	100	4,92	6,59	18,14
3	96:1	644	66	7,38	9,89	27,22
3	144:1	664	44	11,07	14,83	30,00
3	216:1	666	30	16,61	22,25	"
4	256:1	4444	25	19,69	26,37	"
4	384:1	6444	17	29,53	30,00	"
4	576:1	6644	11	30,00	"	"
4	864:1	6664	7,5	"	"	"
5	1024 :1	44444	6,0	"	"	"
5	1536 :1	64444	4,0	"	"	"
5	2304 :1	66444	2,7	"	"	"
5	3456 :1	66644	1,8	"	"	"
5	5184 :1	66664	1,2	"	"	"
5	7776 :1	66666	0,8	"	"	"

Tabla N°8.4 – Características del motor MR83-1,2.

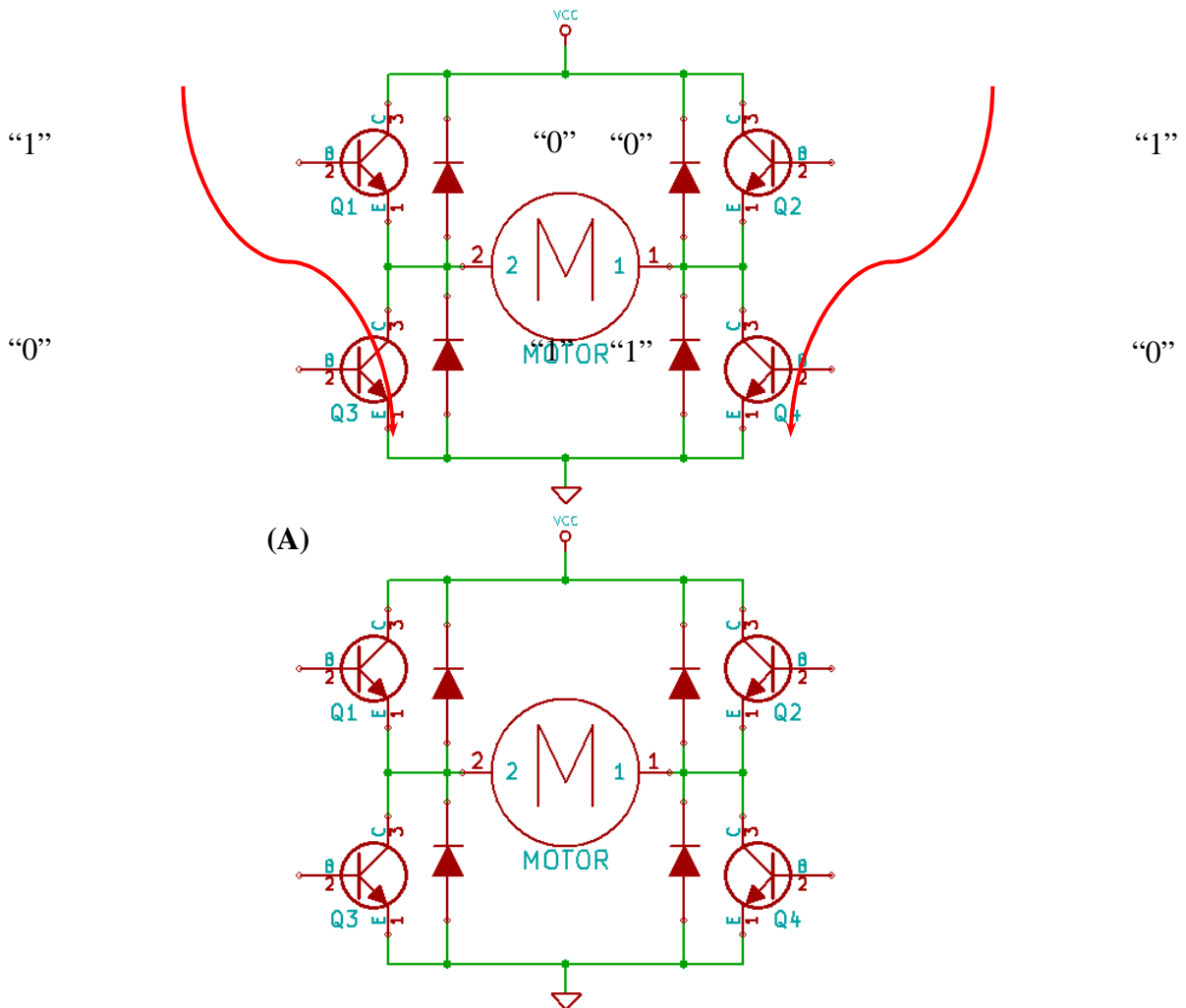
### 8.8.1 - Controlador de motores

Para mover los motores se puede utilizar un circuito denominado puente H, el cual se emplea en motores de corriente continua para permitir el giro en ambos sentidos. Se denomina puente H por la representación gráfica del circuito.

Este circuito controla el sentido de giro mediante el control de cuatro interruptores lógicos, los cuales hacen que la corriente fluya a través del motor en un sentido o el otro como se ve en la figura N°8.13.

Al aplicar un “1” lógico en la base del transistor Q1 y Q4 y un “0” lógico en la base de Q2 y Q3, la corriente fluye en el sentido que se observa en la figura 8.3-A.

Al aplicar un “1” lógico en la base del transistor Q2 y Q3 y un “0” lógico en la base de Q1 y Q4, la corriente fluye en el sentido que se observa en la figura 8.3-B.



**Figura 8.13 - Estado del puente H.**

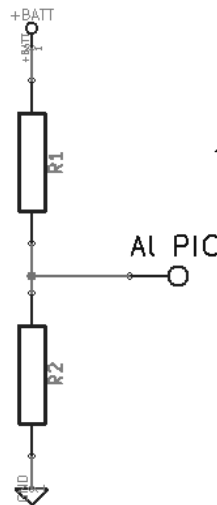
Los puentes H pueden ser encontrados en circuitos integrados, lo que facilita la implementación y reduce tanto el costo como el tamaño del circuito. En este proyecto utilizaremos el integrado L298N, el cual contiene dos puente H permitiéndonos manejar dos motores con el mismo integrado. El L298N puede ser alimentado con una tensión de hasta 50 V y proveer una corriente máxima de 2<sup>a</sup> por salida.

## 8.9 - Medición de tensión, posición vertical y posición horizontal

### 8.9.1 - Medición de tensión

Para medir esta magnitud utilizaremos el convertor analógico digital del microcontrolador por lo tanto la batería máxima a muestrear es de 5V pero nuestro panel puede generar hasta 17V por lo que se diseñara un divisor resistivo (Figura N°8.13) tal como se indica con la fórmula N° 8.14:

$$5V = 17V \cdot \frac{R2}{R1 + R2} \quad (8.14)$$



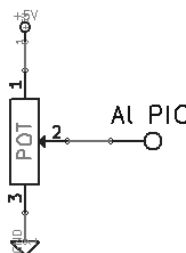
**Figura 8.1 - Divisor resistivo para la medición de tensión de la 189atería.**

Se ha seleccionado  $R2 = 3,3K\Omega$ , entonces  $R1 = 7,92 K\Omega$

Por lo que adoptamos  $R1 = 8,2 K$

### 8.9.2- Medición de posición vertical y horizontal

La posición vertical y horizontal viene dado por el ángulo cenital y azimutal respectivamente para sensar dichas posiciones se volverá a hacer uso del conversor analógico digital del microcontrolador, pero en este el divisor resistivo estará compuesta por un potenciómetro multivuelta cuya resistencia varía con la posición angular ,figura N° 8.15:



**Figura N°8.15 - Medición de posición angular**

### 8.10 - Soporte mecánico del panel solar

En la figura N°8.16 se presentan dos vistas de la estructura mecánica del soporte del panel solar junto con sus medidas en milímetros.

Los sensores están montados a los lados del panel como se ve en la figura N° 8.17.

En la figura N° 8.18 se presenta una foto del soporte terminado.

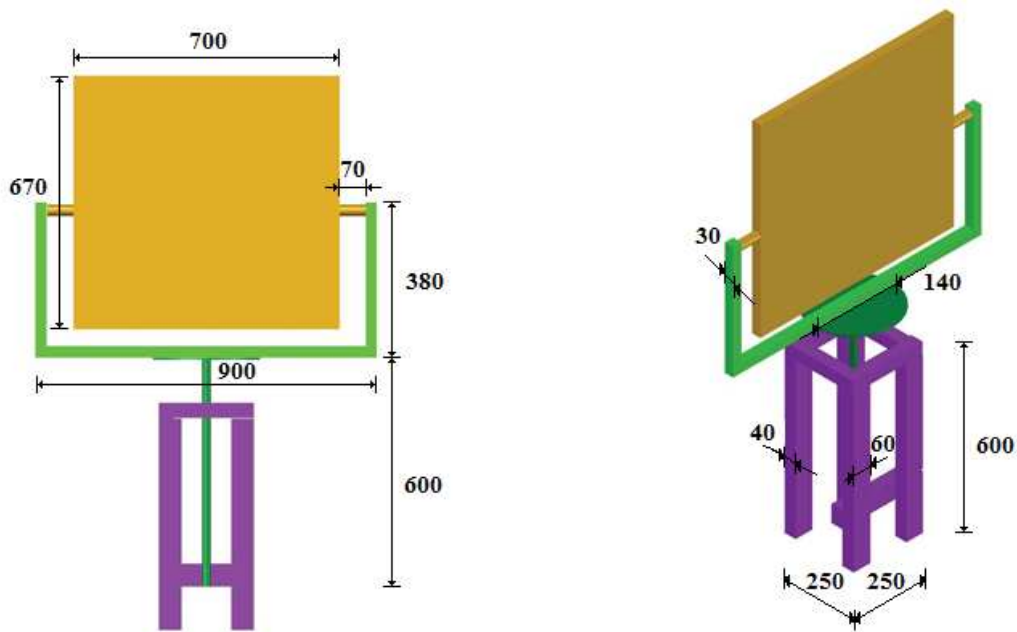


Figura N°8. 16 - Medidas del sistema.

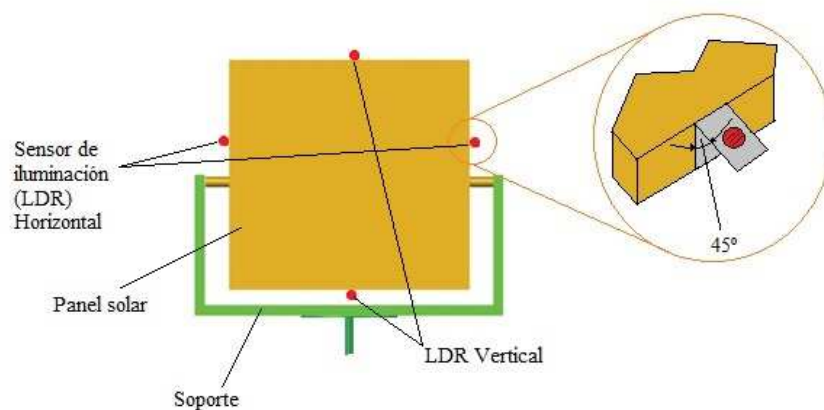


Figura N°8.17 - Ubicación de los sensores LDR

### 8.11 – Firmware

El firmware que fue desarrollado con el software MPLAB IDE de Microchips se muestra en el diagrama de flujo observa en figura N° 8.19 ,donde se observa en el comienzo la inicialización de las variables, del módulo conversor analógico digital (ADC) y del módulo de comunicación serial.

Luego se muestrea el valor de tensión ( $V_f$ ) existente entre los LDRs verticales y se compara con la mitad de la tensión de alimentación que es de aproximadamente 5V aunque en realidad se ha comparado contra un rango de valores en el que está incluido  $V_{DD}/2$ , esto se debe a las tolerancias inherentes al sistema mecánico y la posición de los LDRs.

Si el valor muestreado cae dentro del rango antes mencionado el motor vertical se detiene, por el contrario si no se contiene al valor, se verifica si este es menor o mayor a  $V_{DD}/2$  y se decide si el motor girará en sentido horario o antihorario

respectivamente.

Una vez que se toma una decisión respecto al motor vertical se muestrea el valor de la tensión de los LDRs horizontales y se repite el procedimiento efectuado en el motor vertical para con el motor horizontal, para decidir la condición final del mismo.



**Figura N°8.18 – Soporte terminado**

Por último se muestrea la tensión de la batería, la posición del ángulo cenital y la del ángulo azimutal y se envían, a través del módulo de comunicación serie, hacia una PC para su posterior procesamiento.

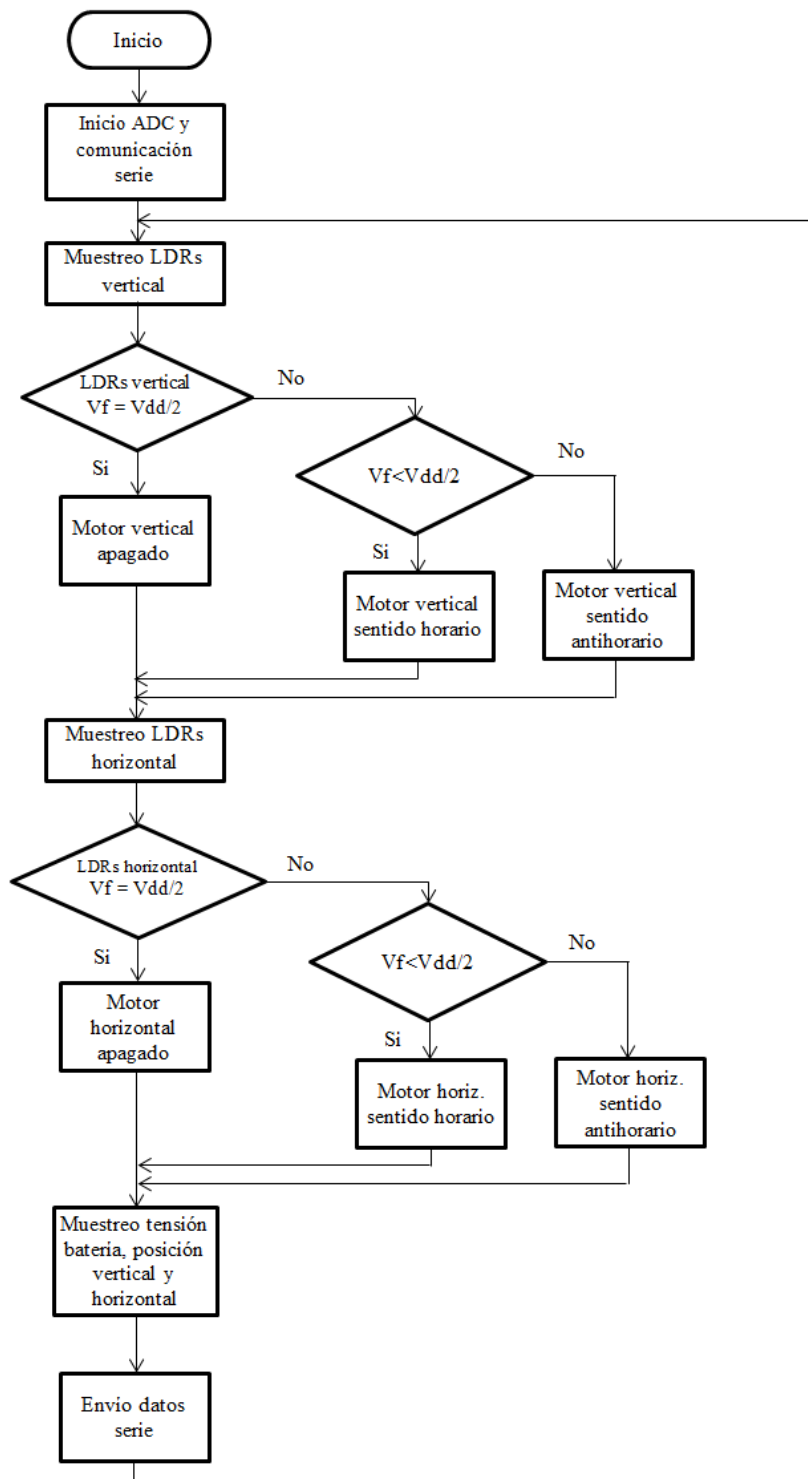
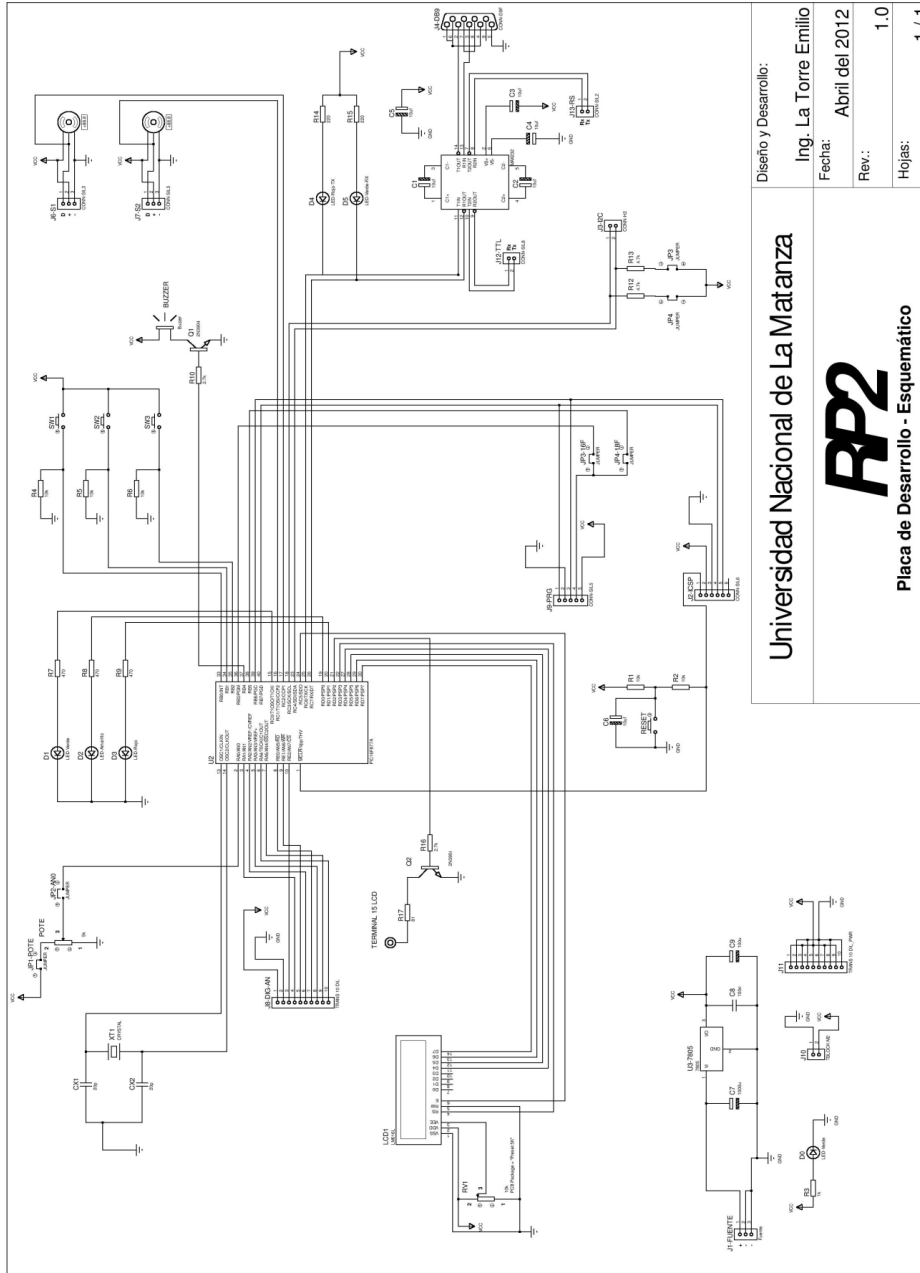


Figura N°8.19 - Diagrama de flujo del firmware



## 8.12- Esquemáticos y PCB

### 8.12.1- Circuito placa microcontrolado



<b>Universidad Nacional de La Matanza</b> <b>RP2</b> <b>Placa de Desarrollo - Esquemático</b>	Diseño y Desarrollo:	Ing. La Torre Emilio
	Fecha:	Abril del 2012
	Rev.:	1.0
	Hojas:	1 / 1

Figura N°8.20 - Esquemático placa de desarrollo.

### 8.12.2-Circuito puente H

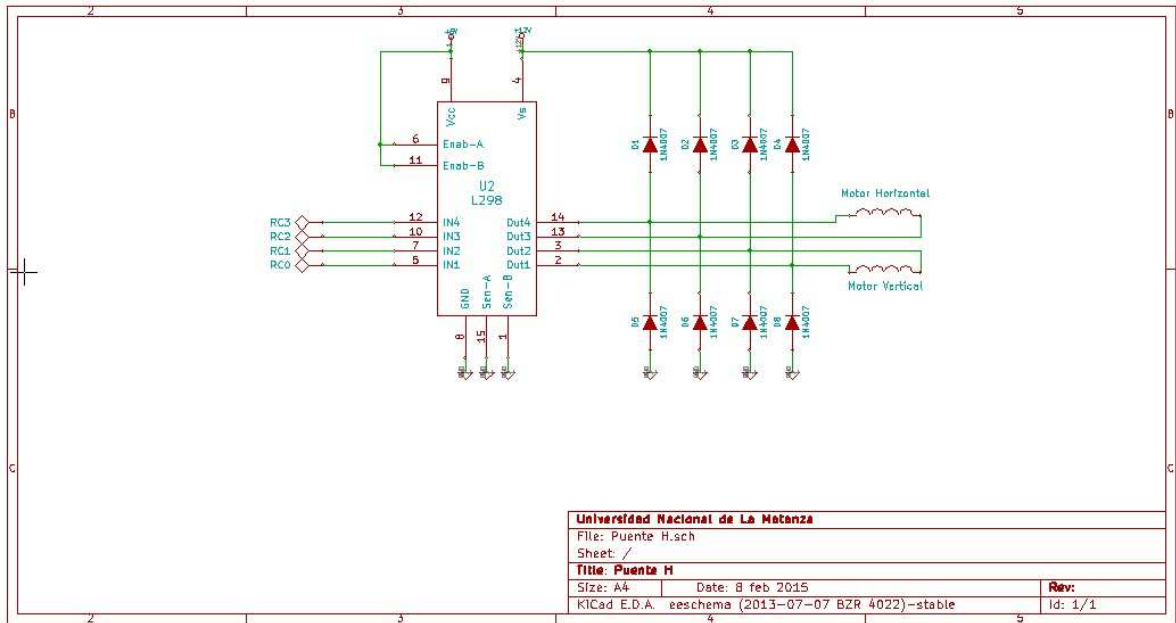


Figura N°8.21 - Esquemático utilizado Puente H.

### 8.12.3- PCB placa micro

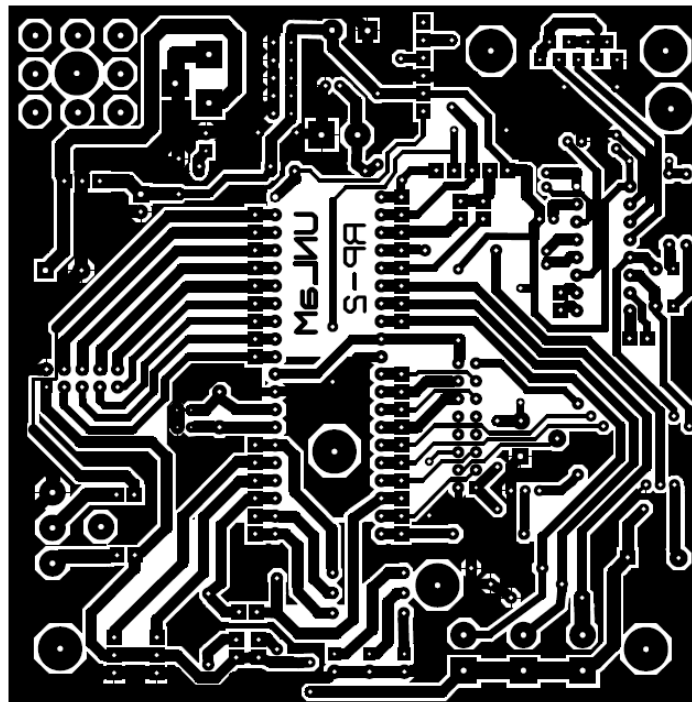


Figura N° 8.22 - Placa de desarrollo para PIC

#### 8.12.4- PCB del puente H

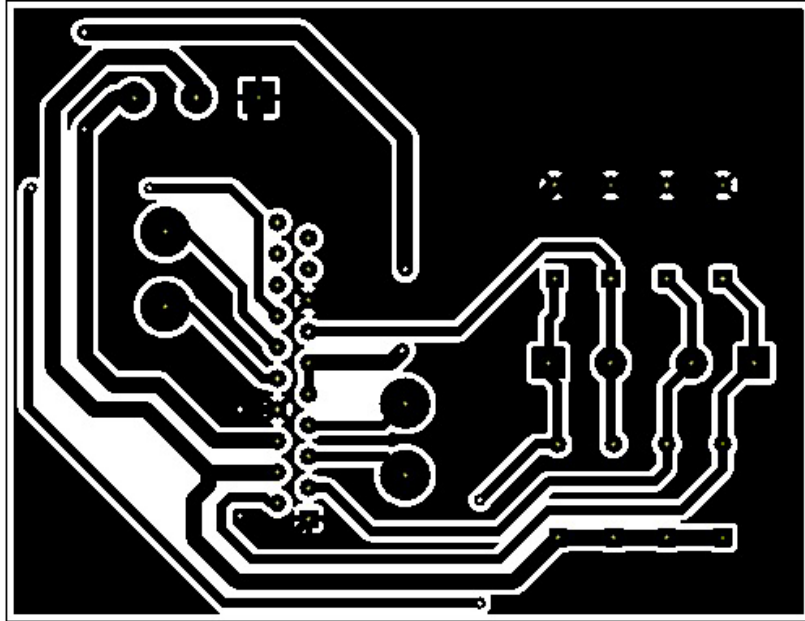


Figura N° 8.23 - PCB puente H

#### 8.13- Evaluación

##### 8.13.1 - Eficiencia

Para medir la eficiencia del proyecto se estimó medir la potencia entregada por el panel a una carga resistiva con el seguidor solar y sin este, es decir con el soporte fijo. En la figura N° 8.24 se presenta el esquema de conexión para la medición:

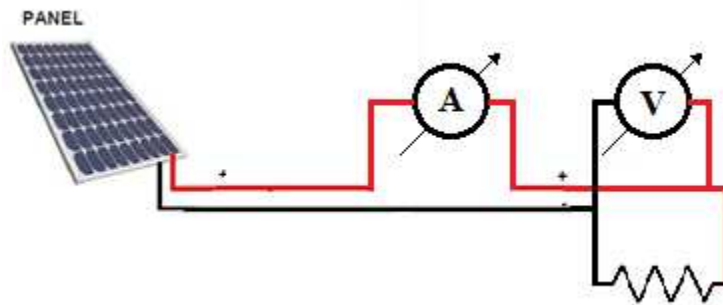
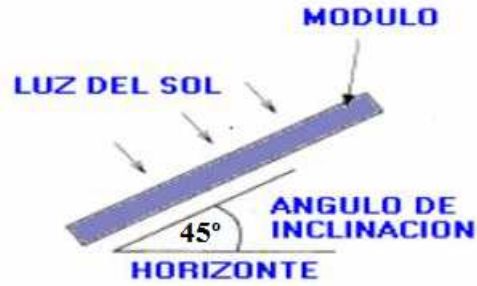


Figura N° 8.24 - Esquema de conexión para medición de carga de la batería.

##### 8.13.2 - Sistema sin seguidor (fijo):

En el sistema sin seguidor solar se desconecta el sistema de control y se orienta el panel hacia el norte geográfico con una inclinación de aproximadamente  $45^\circ$  con respecto a la horizontal (Figura N° 8.25).



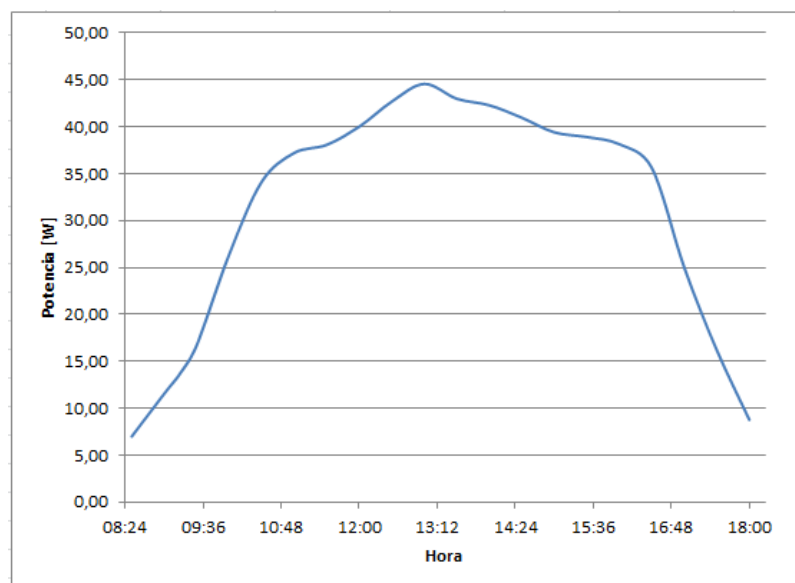
**Figura N° 8.25 - Angulo de inclinación.**

Este ángulo surge de tener en cuenta que la latitud en que se encuentra la ciudad de San Justo es de 34° 41' (obtenida a través de Google Earth), siguiendo los datos de la tabla N°8.5 [63][64]:

Latitud del lugar	Angulo de inclinación fijo
0° a 15°	15°
15° a 25°	La misma latitud
25° a 30°	Latitud más 5 °
30° a 35°	Latitud más 10 °
35° a 40°	Latitud más 15 °
40° o más	Latitud más 20 °

**Tabla N°8.5- Calculo del ángulo de inclinación.**

Una vez orientado el panel, se procede a medir la potencia generada por el panel desde las 08:00hs hasta las 18:00hs, se obtuvieron los datos presentados en el figura N° 8.26:



**Figura N° 8.26 - Potencia de panel solar sin sistema de seguimiento.**

Como se ve tanto en el amanecer como en el atardecer la energía generada disminuye significativamente ya que la intensidad de los rayos solares es menor y a su vez, estos no inciden perpendicularmente con el panel.

Durante el periodo de medición la energía generada fue aproximadamente 337 Wh. Cabe aclarar que las cifras pueden sufrir un leve incremento ya que durante las pruebas hubo lapsos de con cierta nubosidad.

### 8.13.3 - Sistema con seguidor:

Al día siguiente, con similares características climáticas al anterior, se instaló el panel en una posición arbitraria y se encendió el sistema de control, monitoreándose la potencia sobre la carga desde las 08:00 hs hasta las 18:00 hs (Figura N° 8.27)

Durante las horas de mañana la potencia sobre la carga ascendió uniformemente, tendiente al pico máximo (12:00 hora solar) pero en horas del mediodía la curva perdió uniformidad y se tornó inestable debido a dos fenómenos: la presencia de nubes aisladas que entre las 16:00 y 17:00 horas cubrieron en su totalidad al sol (ver mínimo 16:30hs), y la elevada temperatura que tomó el panel, siendo que la temperatura ambiente promedio durante ese período fue de 32°C. Al estar el cielo parcialmente nublado, la temperatura del panel variaba intermitentemente, motivo por el que la potencia sobre la carga variaba de la misma manera.

Durante el periodo de medición la energía generada fue aproximadamente 422 Wh.

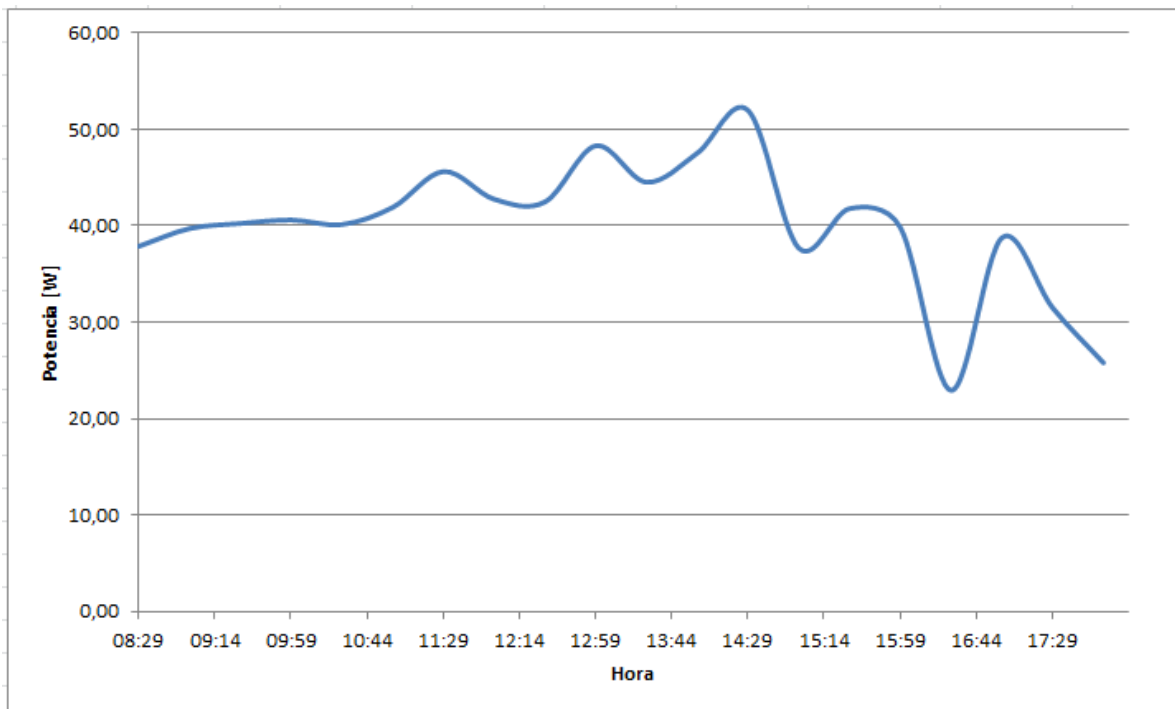
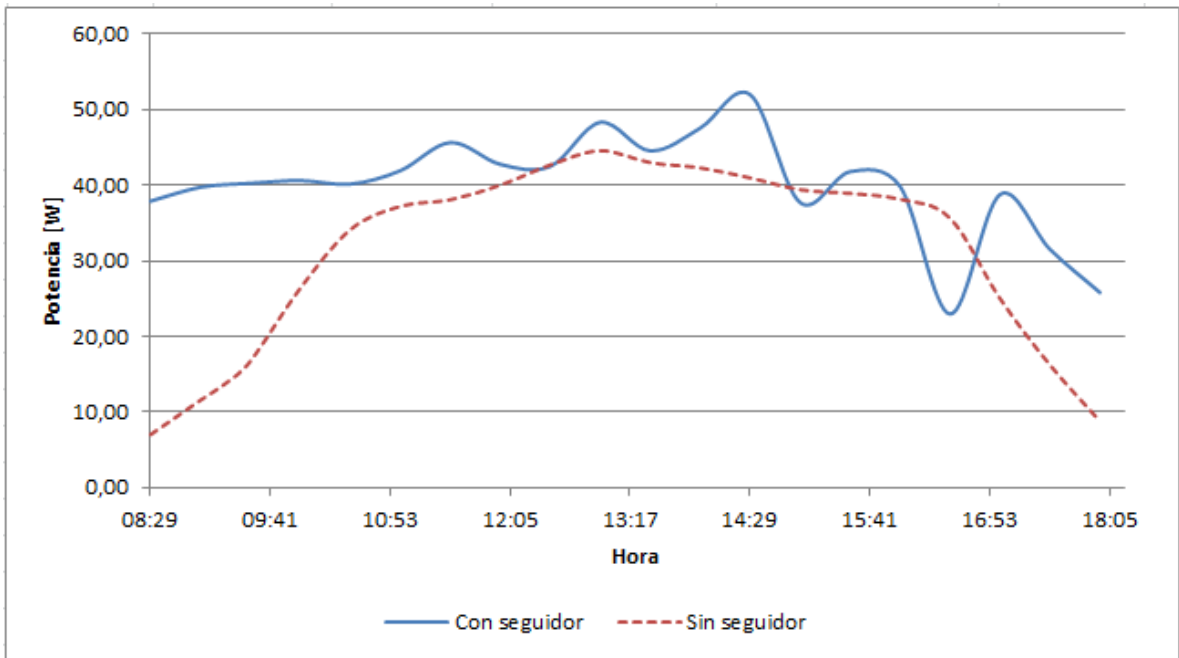


Figura N° 8.27 - Potencia de panel solar con sistema de seguimiento.

### 8.13.4 - Comparación entre sistemas:

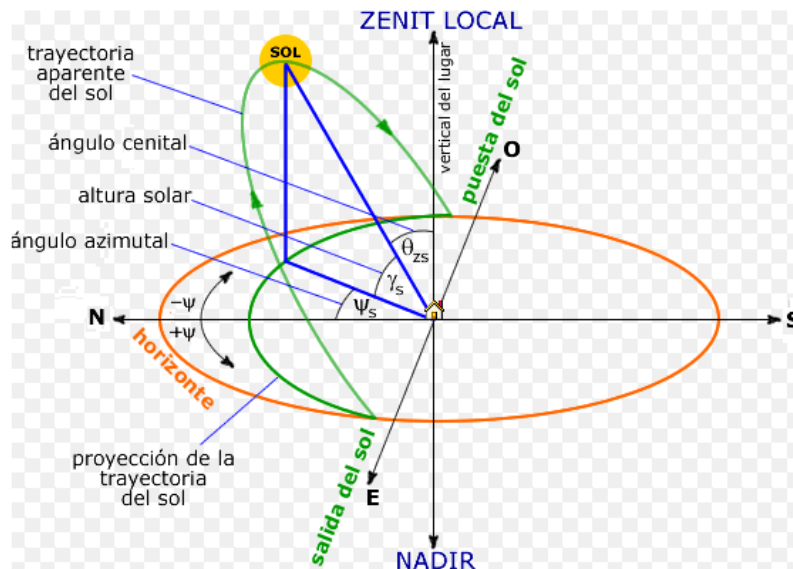
Se puede observar en la figura N°8.28 que la energía generada por el panel con el seguidor es mayor que la de soporte fijo, excepto en horas de la tarde donde, como se detalló anteriormente, había abundante nubosidad, disminuyendo la potencia del sistema con seguidor. En función de las pruebas realizadas se puede asegurar que el sistema permite obtener un aumento de generación de energía de aproximadamente un 25%.



**Figura N°8.28- Comparación de potencia de panel solar con y sin sistema de seguimiento.**

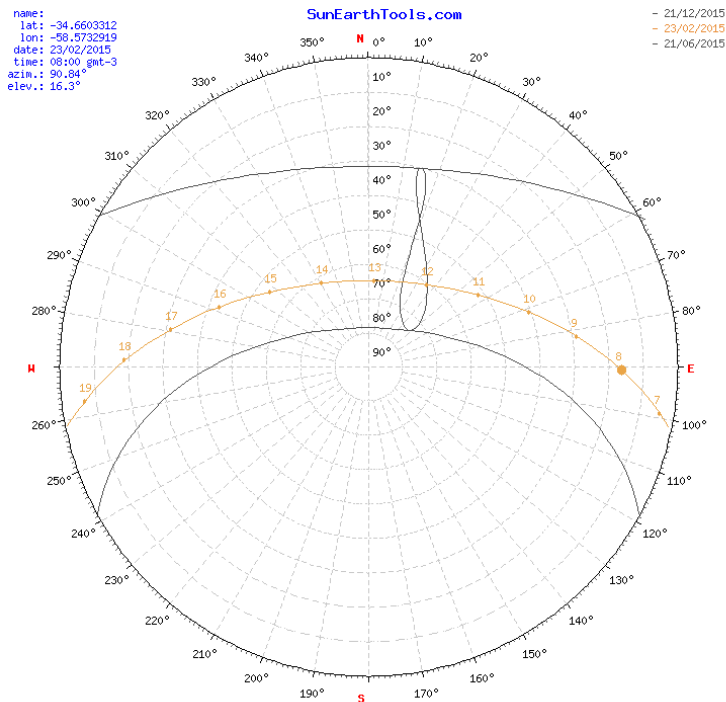
### 8.13.5 - Precisión

Para verificar la precisión del seguimiento se debe conocer la precisión de los ángulos cenital y azimutal (Figura N° 8.29).



**Figura N° 8.29 - Descripción ángulo cenital y azimutal.**

Para conocer la ubicación exacta del sol durante el día se tomó los datos brindados en la página de internet <http://www.sunearthtools.com/>, donde ingresando las coordenadas donde se encuentra el panel, la zona horaria y la fecha se conoce de forma precisa la combinación de ángulo cenital y azimutal que tiene el sol (Figura N°8.30).



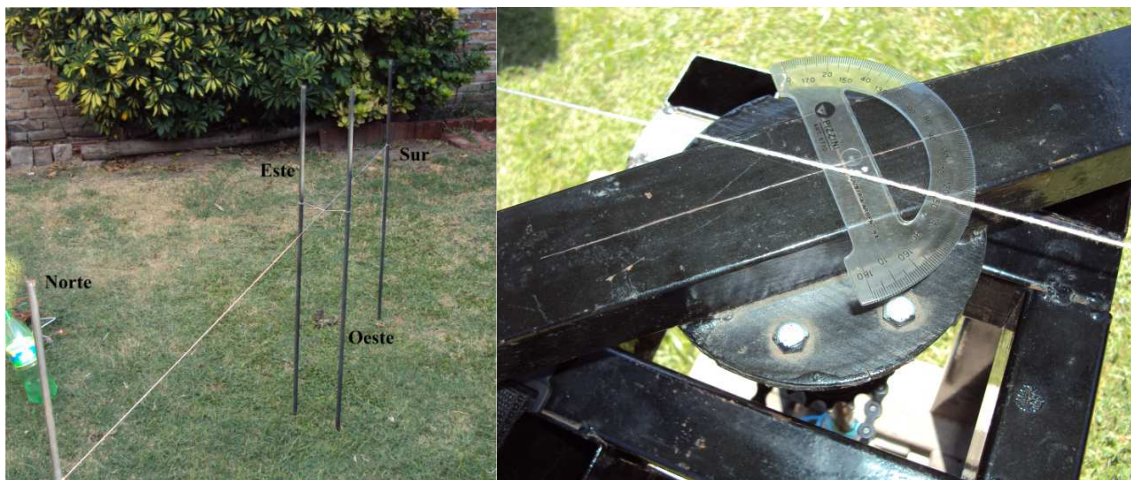
**Figura N°8.30 - Angulo cenital y azimutal del recorrido del solar.**

Teniendo esta información se procede a medir los ángulos para verificar el error cometido en el seguimiento. Para esto se nivelo el terreno y con ayuda de una brújula se marcó la línea Norte-Sur como se observa en la figura N°8.31a.

Con esta referencia se procede a medir los ángulos azimutales (Figura N° 8.31b), mientras que los cenitales se miden por intermedio de una varilla recta que coincide con la vertical del lugar (normal al terreno).

Realizada la experiencia desde la 08:30 hs hasta las 18:00hs, se obtuvieron los datos de la tabla N° 8.6:

Comparando ambos ángulos con su posición real nos queda se obtuvieron los gráficos de las figuras N° 8.32 y 8.33.



**Figura 8.31 - (a) Referencia Norte-Sur, (b)Medición ángulo azimutal.**

Hora	Teorico		Practico		Observaciones
	Cenital [°]	Azimutal [°]	Cenital [°]	Azimutal [°]	
08:30	22,46	86,51	30,00	88,00	despejado
09:00	28,46	81,94	35,00	84,50	despejado
09:30	33,66	76,97	38,00	80,00	despejado
10:00	40,59	71,39	50,00	78,00	despejado
10:30	46,31	64,94	52,00	66,50	despejado
11:00	51,71	57,22	60,00	60,00	despejado
11:30	56,6	47,74	65,00	49,00	despejado
12:00	60,72	35,9	68,00	38,00	despejado
12:30	63,68	21,33	70,00	18,00	Algo nublado
13:00	65,06	4,49	73,00	0,00	Algo nublado
13:30	64,59	347,09	71,00	342,00	Algo nublado
14:00	62,38	331,24	68,00	323,00	despejado
14:30	58,79	318,04	65,00	308,00	despejado
15:00	54,24	307,44	63,00	290,00	Mayormente nublado
15:30	49,07	299,9	59,00	287,00	Mayormente nublado
16:00	43,49	291,87	55,00	280,00	Mayormente nublado
16:30	37,65	285,9	55,00	280,00	Nublado
17:00	31,64	280,66	43,00	280,00	Mayormente nublado
17:30	25,53	275,91	39,00	280,00	Algo nublado
18:00	19,37	271,48	30,00	275,00	Algo nublado

Tabla N° 8.6 - Posición real y práctica del sol.

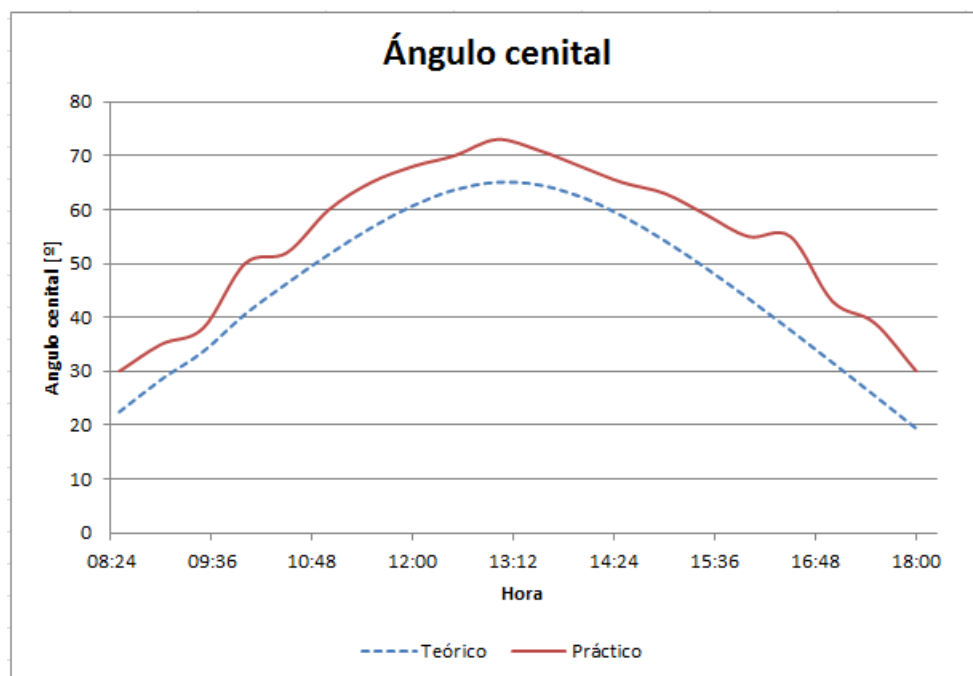
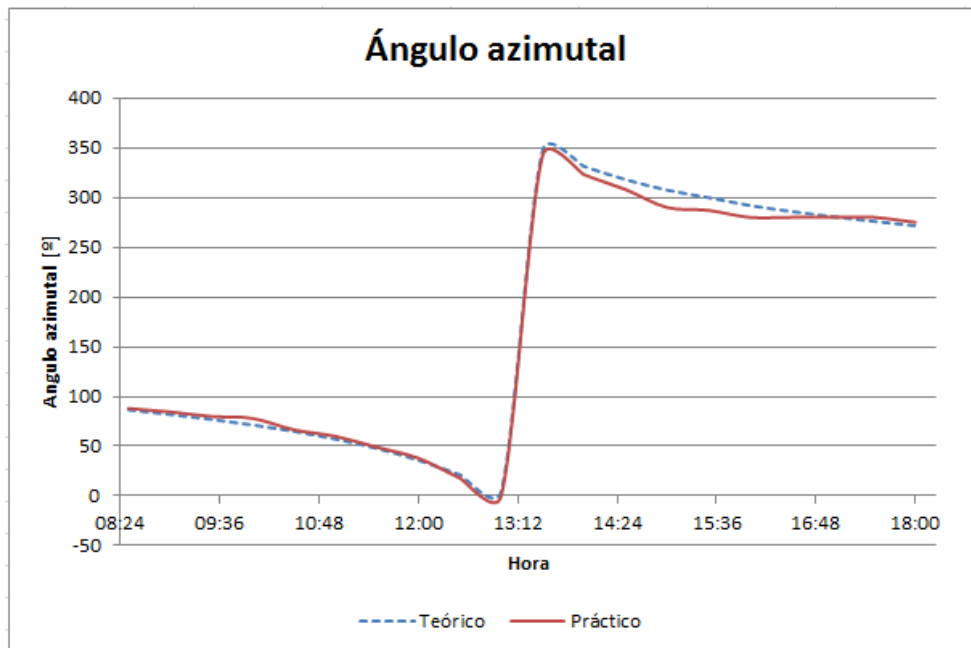


Figura N° 8.32- Comparación entre ángulo cenital teórico y práctico.





**Figura N° 8.33 - Comparación entre ángulo azimutal teórico y práctico.**

Observando la gráfica de la figura N° 8.32 se puede ver que los valores prácticos del ángulo cenital acompañan en forma a los datos teóricos pero no en magnitud ya que, como se observa, posee un desplazamiento ascendente del orden de los  $10^{\circ}$  aproximadamente. Las causas son básicamente dos, primero que se posee un juego considerable en el sistema de transmisión por cadena del movimiento angular y segundo que los sensores tienen orientación ligeramente diferente a  $45^{\circ}$ . Cabe recordar que como se ve en la Tabla N° 8.6 y en la inestabilidad de la gráfica luego de las 15hs y hasta las 17hs hubo presencia de nubosidad.

Analizando el gráfico de la figura N° 8.33 del ángulo azimutal, y la tabla N°8.6, se observa que presenta un error angular promedio de  $4^{\circ}$  por las mismas causas del error del ángulo cenital, en el horario donde hay presencia de nube este error aumenta a más de  $10^{\circ}$ , por lo que se puede decir que el seguidor, en este sentido posee cierta precisión.

#### 8.14- Conclusiones

Se logró el desarrollo de un sistema fotovoltaico con seguimiento de dos ejes, con una eficiencia de aproximadamente 25% mayor que un sistema fotovoltaico fijo.

Los valores de potencia que se midieron sobre la carga del sistema con seguimiento solar, permitieron verificar los cálculos previamente realizados.

Con respecto a la exactitud del sistema se determinó que el ángulo azimutal presenta un error promedio de  $4^{\circ}$ , que es un error aceptable, en cambio para el ángulo cenital se encontró que contiene un desplazamiento ascendente de  $10^{\circ}$  debiéndose ajustar a futuro el mecanismo de transmisión del movimiento angular o reconsiderar el tipo de mecanismo por otro que presente menos error.

La utilización de los sensores LDR, levemente desestimados por su alinealidad, tuvieron un desempeño de funcionamiento correcto.

En el futuro se esperan numerosas mejoras, desde aspectos mecánicos tendientes

a mejorar la exactitud y reducir los costos, aspectos de control con la implementación de un control híbrido basado en sensores y calendario astronómico, hasta el desarrollo de una aplicación remota con el objeto de monitorear los parámetros del sistema para hacer uso más eficiente de los recursos.

## 9 - Biodigestor

### 9.1 - Introducción

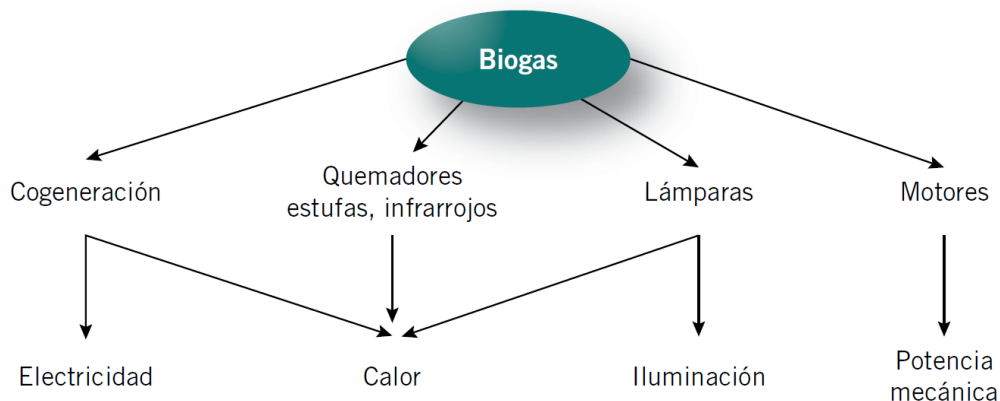
El uso de energías no renovables como el carbón, petróleo, gas natural y combustibles nucleares tiene una serie de consecuencias negativas para la humanidad. Es por eso que desde hace varios años diversos organismos tanto públicos como privados se encuentran en la búsqueda de optimizar el uso de las energías renovables, de modo de lograr una competitividad que la coloque entre las alternativas viables para las grandes industrias.

Entre las distintas tecnologías alternativas para la producción de energía se encuentra el biodigestor, que por sus características podría ser implementado a pequeña escala (familiar) logrando en principio complementar, cuando no reemplazar, al consumo de gas desde la red.

El biogás es el producto de la conversión bioquímica o digestión de biomasa orgánica. Se llama biogás a la mezcla constituida por metano ( $\text{CH}_4$ ) en una proporción que oscila entre un 50% y un 70% y dióxido de carbono que contiene pequeñas proporciones de otros gases tales como hidrógeno, nitrógeno y sulfuro de hidrógeno. El proceso además puede generar biofertilizante, lo que lo convierte en una forma atractiva de generar energía y abono a partir de desechos orgánicos como los que produce la agricultura.

Un biodigestor es un dispositivo capaz de generar estos productos a partir de desechos orgánicos, siendo el gas utilizado como combustible en cocinas, calefacción, iluminación y el abono utilizado como fertilizante ecológico en todo tipo de cultivos [64].

En la figura N° 9.1 se ven los posibles abastecimientos de energía que se pueden lograr con biogás.



**Figura N° 9.1- Los posibles abastecimientos de energía que se puede lograr con el biogás.**

En la tabla N°9.1 se muestra el consumo promedio de algunos artefactos que emplean biogás. Las cocinas y calentadores son fácilmente modificables, agrandando el paso del gas de los quemadores. La amplia disponibilidad de este tipo de equipos hace promisorio e interesante su utilización a gran escala.

Las lámparas a gas tienen una muy baja eficiencia y el ambiente donde se las utilice debe estar adecuadamente ventilado para disipar el calor que generan.

ARTEFACTO	CONSUMO
Quemador de cocina	300 - 600 l/h
Lámpara a mantilla (60 W)	120 - 170 l/h
Heladera de 100L	30 - 75 l/KWh
Motor a gas	0.5 m <sup>3</sup> /KWh
Quemador de 10 KW	2 m <sup>3</sup> /h
Infrarrojo de 200 W	30 l/h

**Tabla N°9.1 - Artefactos que utilizan biogás y sus respectivos consumos.**

Las heladeras domésticas constituyen un interesante campo de aplicación directo del biogás debido a que tienen un consumo parejo y distribuido a lo largo de las 24 horas del día lo cual minimiza la necesidad de almacenaje del gas. Estos equipos funcionan bajo el principio de la absorción (generalmente de ciclo amoníaco refrigerante - agua absorbente).

Los quemadores infrarrojos comúnmente utilizados en la calefacción de ambientes (especialmente en criadores) presentan como ventaja su alta eficiencia lo cual minimiza el consumo de gas para un determinado requerimiento térmico.

El biogás puede ser utilizado en motores de combustión interna tanto nafteros como diesel. El gas obtenido por fermentación tiene un octanaje que oscila entre 100 y 110 lo cual lo hace muy adecuado para su uso en motores de alta relación volumétrica de compresión, por otro lado una desventaja es su baja velocidad de encendido.

En los motores de Ciclo Otto el carburador convencional es reemplazado por un mezclador de gases. Estos motores son arrancados con nafta y luego siguen funcionando con un 100% de biogás con una merma de la potencia máxima del 20% al 30%.

A los motores de Ciclo Diesel se les agrega un mezclador de gases con un sistema de control manteniendo el sistema de inyección convencional. De esta manera estos motores pueden funcionar con distintas proporciones de biogás-diesel y pueden convertirse fácil y rápidamente de un combustible a otro lo cual los hace muy confiables. [66]

Beneficios:

- Mejora la salud y la ecología de las comunidades rurales al evitar el uso de leña para cocinar.
- Promueve la conservación de áreas naturales y bosques.
- Reduce el uso de costosos y contaminantes fertilizantes químicos.
- Al disponer adecuadamente del estiércol de los animales, se minimizan los riesgos para la salud.
- Reducción de emisiones de gases invernadero que contribuyen al calentamiento global.

## **9.2 - Biodigestores en Argentina**

Argentina se caracteriza por un sólido sector agropecuario y agroindustrial, producción de granos, carnes, productos lácteos, alimentos, etc. Esta actividad genera

una gran cantidad y diversidad de residuos y subproductos agropecuarios, como por ejemplo residuos de procesos agroindustriales y estiércoles (materia fecal animal y agua).

La FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) [67] es una organización que está conformada por las siguientes instituciones:

- La Secretaría de Energía
- La Secretaría de Ambiente y Desarrollo Sustentable
- La Secretaría de Agricultura, Ganadería, Pesca y Alimentación
- El INTA

Esta organización ha llevado adelante una evaluación detallada de la biomasa natural, utilizando el modelo WISDOM. Esta evaluación contempla una estimación realizada sobre la base de información secundaria proveniente de estadísticas productivas con datos sobre la distribución geográfica del potencial de la leña, otra biomasa leñosa (generada por el procesamiento de los rollizos), el bagazo, residuos agrícolas y agroindustriales. El estudio del proyecto de FAO indica que la biomasa total accesible y potencialmente contabilizada por el modelo utilizado ascendió a más de 148 millones de toneladas, en tanto que la denominada biomasa comercial accesible y potencialmente disponible alcanzó más de 124 millones de toneladas [68].

Tras la gran expansión del biodiesel y del bioetanol, el biogás busca dentro de los biocombustibles un lugar en la Argentina. El biogás se instaló en Europa para agregar valor a los cultivos energéticos, los efluentes agropecuarios, los residuos sólidos urbanos y los residuos orgánicos industriales como los de los rellenos sanitarios. Ahora, está insertándose lentamente en la Argentina.

El funcionamiento exitoso del primer biodigestor de la Argentina – desarrollado por investigadores de la Universidad Nacional del Litoral – contribuye al debate sobre la basura domiciliaria en el país. La instalación permite procesar los residuos orgánicos de la localidad santafesina de Emilia – de unos mil habitantes: la basura se transforma en abono orgánico y en gas metano utilizable como combustible. El único insumo que requiere su puesta en marcha es un poco de estiércol de vaca, para proveer las bacterias que efectúan el proceso. Internacionalmente, los biodigestores recibieron gran impulso en Europa durante la última década, y hoy procesan gran parte de la basura domiciliaria de ciudades como Barcelona y Verona. Pero su efectividad también se demuestra en conglomerados latinoamericanos como Río de Janeiro. El procesamiento racional de los residuos no sólo resuelve la presencia de los basurales, sino que evita la liberación de gases tóxicos a la atmósfera e incluso reduce, en las rutas, los accidentes causados por el humo procedente de los basurales.

### **9.3 - Introducción teórica**

#### **9.3.1 - El proceso**

Los agentes de este proceso son bacterias anaerobias, es decir, se desarrollan en ambientes libres de oxígeno.

Las bacterias fermentan el material orgánico en ausencia de aire (es decir, fermentación anaeróbica) y producen biogás. Este material de fermentación está constituido por sustancias sólidas orgánicas, inorgánicas y agua (el cual incrementa la fluidez del material de fermentación, característica importante para el funcionamiento de la planta de biogás). El biogás que generalmente se produce es un gas incoloro, inflamable y contiene 60% de metano y 40% de dióxido de carbono, con aportes

menores de Nitrógeno, Hidrógeno y gas sulfhídrico. Su poder calorífico es de 4400 Kcal/m<sup>3</sup> [69].

El proceso de fermentación se compone de tres fases principales:

- Primera fase: de hidrólisis, donde las bacterias fermentativas o acidogénicas hidrolizan los polímeros y las convierten a través de la fermentación en ácidos orgánicos solubles.
- Segunda fase: de acidificación, donde las bacterias acetogénicas causan una metabolización de los ácidos orgánicos en acetatos (CH<sub>3</sub>COOH), dihidrógenos (H<sub>2</sub>) y carbodióxidos (CO<sub>2</sub>).
- Tercera fase: de metanización, donde las proteínas, hidratos de carbono y grasa, los aminoácidos, alcoholes y ácidos grasos que se formaron en las fases anteriores, se convierten en metano, bióxido de carbono y amoníaco.

### 9.3.2 - Factores influyentes

Se mencionaran los factores que influyen en el funcionamiento del biodigestor:

- a. Temperatura y tiempo de retención: el rango de temperatura y el periodo de retención dentro del biodigestor, permiten realizar la siguiente clasificación:
  - *Fermentación psicrófila*; entre 10 y 20°C y más de 40 días de retención.
  - *Fermentación mesófila*; entre 20 y 35°C y entre 10 a 40 días de retención.
  - *Fermentación termófila*; entre 35 y 60°C y aprox. 8 días de retención.
- b. Relación Carbono/Nitrogeno (C/N) de los materiales de fermentación, influye sobre la producción de gas. Una relación de 20:1 hasta 30:1 es aceptable aunque el valor ideal es de 16. En la tabla N° 9.2 se muestran relaciones C/N para varios productos residuales:

Sustancia	Relación C/N
Orina	0.80
Estiércol equino	25
Estiércol vacuno	18
Alfalfa	16-20
Algas marinas	19
Aserrín	511
Basura	25
Cáscaras de papa	25
Paja seca de trigo	87
Paja seca de arroz	67
Tallo del maíz	53
Hojas secas	41
Estiércol de aves	32
Pasto	27
Estiércol ovino	29
Estiércol de cerdos	13
Excretas frescas humanas	2.90

**Tabla N°9.2 - Relaciones C/N de varios productos residuales.**

c. pH: Está en función de la concentración de CO<sub>2</sub> en el gas, de la concentración de ácidos volátiles y de la propia alcalinidad de la materia prima. Las bacterias responsables del mecanismo de producción de biogás son altamente sensibles a cambios en el pH, oscilando entre 6 y 8 (se recomienda entre 7 y 7.2). El pH del lodo de fermentación indica si este proceso transcurre sin problemas, y su medición indica el comportamiento de la carga de fermentación dentro del digestor, para la producción de biogás ,tabla N°9.3:

pH	Comportamiento
7 – 7.2	Óptimo
6.2	Retarda la acidificación
7.6	Retarda la amonización

**Tabla N° 9.3 - Comportamiento de la carga de fermentación dentro del biodigestor de acuerdo con el valor del pH.**

d. Contenido de agua de la mezcla: Las bacterias y otros microorganismos no pueden funcionar efectivamente cuando el contenido de agua de la mezcla es demasiado bajo, la cantidad de biogás producido será pequeña. Cuando la mezcla es demasiado diluida, se puede digerir relativamente poca materia orgánica y la producción del biogás es limitada. El uso primordialmente de excreta humana y orines, estiércol, y desechos de agricultura, como alimento para el digestor, deberá conllevar a una razón de biomasa a agua entre 1:1 y 1:2; y por cada 100 Kg de heces y orina, se requerirán entre 100 y 200 litros de agua. Cuando el material de alimento consta principalmente de residuos vegetales, se requiere más agua, en una razón de 1:3 o 1:4. La actividad de mezclar, debe realizarse en forma adecuada y uniforme en el tanque del digestor para promover una digestión efectiva, especialmente si se utiliza biomasa cruda con alto contenido leñoso.

e. Materiales orgánicos: Las actividades pecuarias y agrícolas, producen materiales orgánicos que pueden ser tratados mediante el proceso anaeróbico. Otros materiales orgánicos de origen animal también pueden emplearse como sustrato para la obtención de biogás y bioabono, cuyas características se muestran en la Tabla N°9.2. Los residuos vegetales, de la Tabla N°9.3, como paja, pasto y desechos de verdura, pueden fermentarse anaeróbicamente debiendo previamente triturarse para evitar la formación de capa flotante y dar un tratamiento especial en plantas donde se hace una sola carga hasta que el material se descompone.

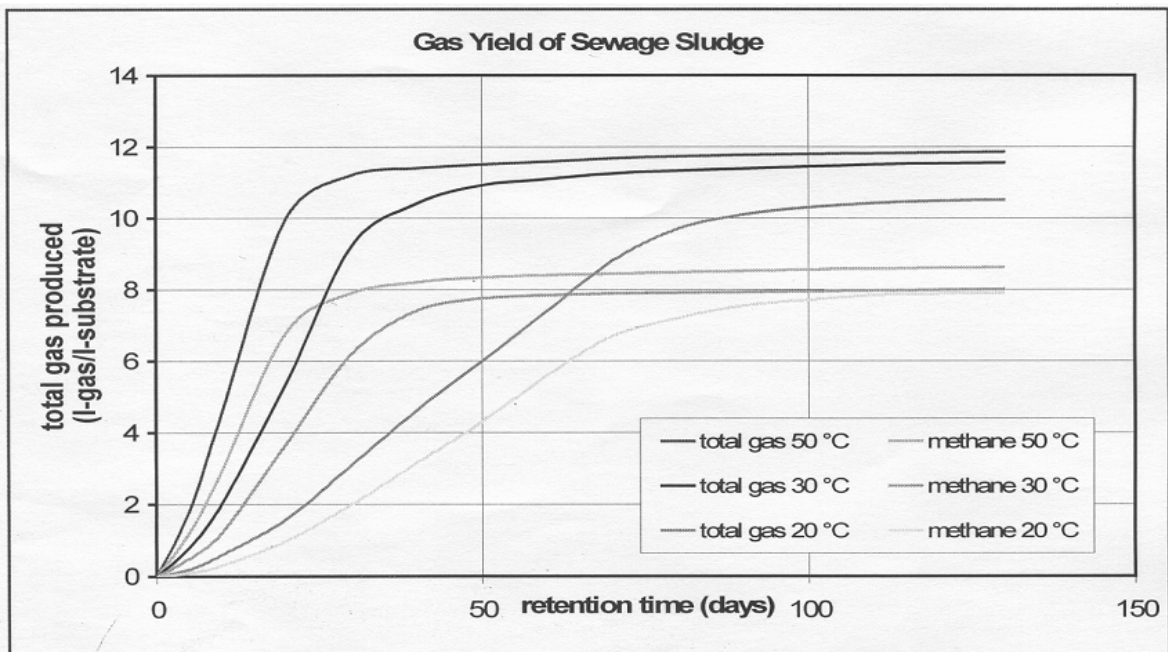
Origen	Cantidad diaria		% material de fermentación		Litros de biogás por un kg de sólidos orgánicos
	Estiércol en % del peso vivo	Orina en % del peso vivo	% Sólidos totales	% Sólidos orgánicos	
Caprinos y ovinos	3	1.5	30	20	200
Equinos	5	4.0	25	15	250
Avícola	4.5	4.5	25	17	400

**Tabla N°9.2 - Características de otros materiales orgánicos de origen animal, utilizados para carga en biodigestores.**

Material fresco	% sólidos totales	% sólidos orgánicos	Litros de biogás por un kg de sólidos orgánicos
Paja de arroz	89	93	220
Paja de trigo	82	94	250
Paja de maíz	80	91	410
Hierba fresca	24	89	410
Jacinto de agua	7	75	325
Bagazo	65	78	160
Desecho de verdura	12	86	350

**Tabla N°9.3 - Residuos vegetales y posible producción de biogás.**

El Dr Schulz H. llevó a cabo un gran número de experiencias sobre la duración de la digestión a diferentes temperaturas, los resultados se muestran en las curvas presentadas en la figura N°9.2:



**Figura N°9.2 - Producción de gas a partir de la degradación de materia orgánica sometida a diferentes temperaturas.**

Se puede apreciar que la productividad de gas metano aumenta con la temperatura, hasta llegar a un valor límite de degradación del material a fermentar.

#### 9.4 - Materiales

Se realizaron diversos estudios en el laboratorio del Institute of Technology and Biosystems Engineering – Germany, en búsqueda de un material que desarrolle el máximo crecimiento de producción de biogás, a una temperatura constante de 35 °C , se estudió la producción de tres digestores de 20 litros de capacidad, los reactores se cargaron con 15 litros de desechos orgánicos.



Los materiales analizados fueron:

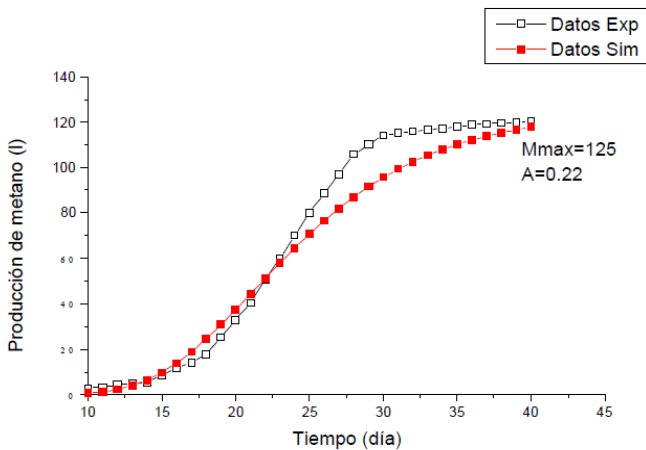
- Excremento de vaca
- Excremento de cerdo
- Maíz

Las curvas de crecimiento observadas experimentalmente (Figura N° 9.3 ,9.4 y 9.5) permitieron comprobar el modelo matemático propuesto por el Dr. Schulz (1996), concluyendo además que la velocidad de generación de gas es mayor al usar estiércol de cerdo, observando un apartamiento del 10% del modelo.

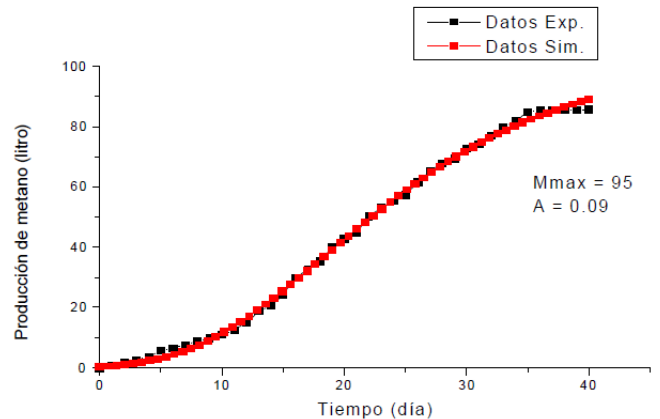
En la tabla N° 9.3 se presentan los resultados experimentales de valor  $M_{max}$ :

	Excremento de vaca	Excremento de cerdo	Mezcla de aceite y grasa
$M_{max}$ Experimental Litros metano/kg sólido volátil	85	110	135
$M_{max}$ De acuerdo con la ecuación (3)	90	110	150
$M_{max}$ De acuerdo con la ecuación (4)	90	111	125

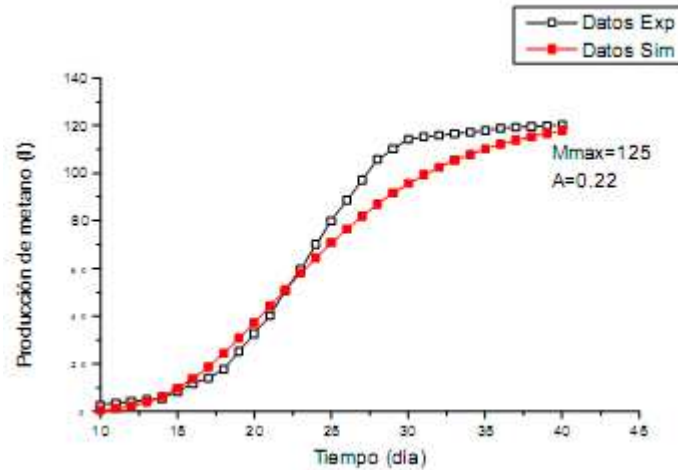
**Tabla N° 9.3 - Máxima generación de metano obtenida en forma experimental y teórica.**



**Figura N°9.3: Producción de metano generado por la fermentación de maíz.**



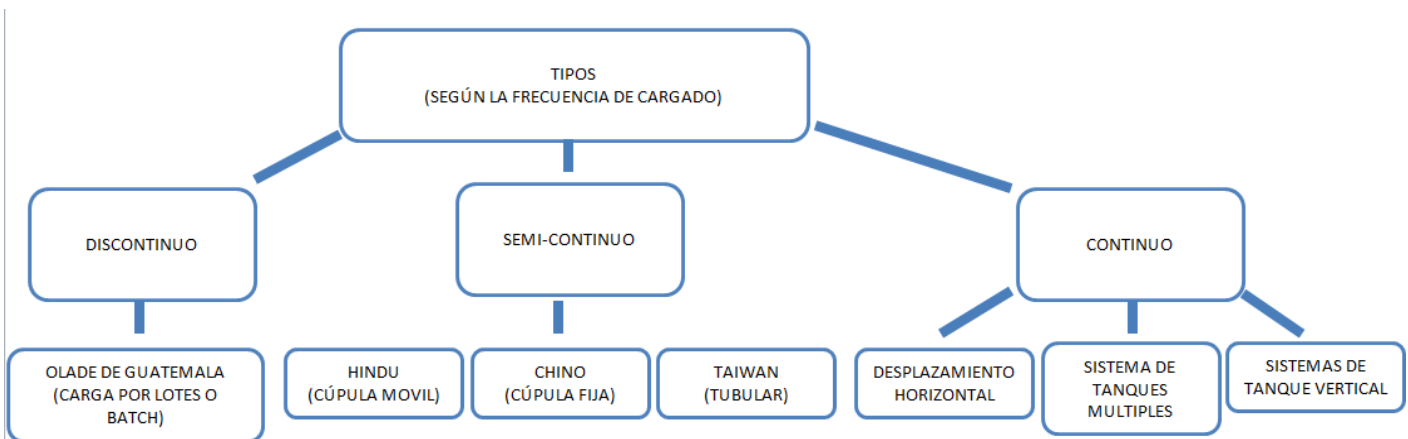
**Figura N°9.4: Producción de metano generado por la fermentación de estiércol de vaca.**



**Figura N°9.5: Producción de metano generado por la fermentación de estiércol de cerdo.**

### 9.5 - Clasificación de Biodigestores

Los biodigestores se clasifican según su forma de operación y su frecuencia de carga. Según su frecuencia de carga podemos encontrar biodigestores de tipo discontinuo, semi-continuo y continuo. Cada uno con sus correspondientes ventajas y desventajas (Figura N°9.6) [70].



**Figura N°9.6 - Clasificación de los biodigestores según la frecuencia de cargado.**

**Discontinuo:** Se cargan una sola vez y se retira cuando ya se ha dejado de producir gas, en ese momento se renueva de materia orgánica. Se usa cuando la disponibilidad de materia orgánica es limitada o intermitente.

**Semi-continuo:** Se cargan en lapsos cortos de 12 horas, 1 vez al día, o cada dos días. Se utilizan cuando la disponibilidad de materia orgánica es constante. Los principales son el Hindu, el Chino y el Taiwanés. Son los más usados en medios rurales.

**Continuo:** Se cargan continuamente y principalmente tienen la finalidad de tratamiento de aguas negras, como por ejemplo los desechos de los baños de un hogar o en industrias.

### 9.5.1 - Biodigestor de carga discontinua o intermitente (Tipo Batch)

Esta clase de biodigestor, se llena una vez y se descarga el contenido digerido, cuando finaliza el proceso de fermentado o sea cuando deja de producir gas. Tiene dos orificios, uno para la carga y otro para la descarga. La duración de la fermentación varía entre 1 a 4 meses, dependiendo del clima cálido, templado, frío, se puede utilizar un mecanismo adicional para el acondicionamiento de la temperatura, ya que esta afecta directamente la velocidad de reacción dentro del reactor (Figura N° 9.7). Este sistema tiene a la materia confinada del inicio al fin del proceso, no hay sostenibilidad en la producción de biogás.

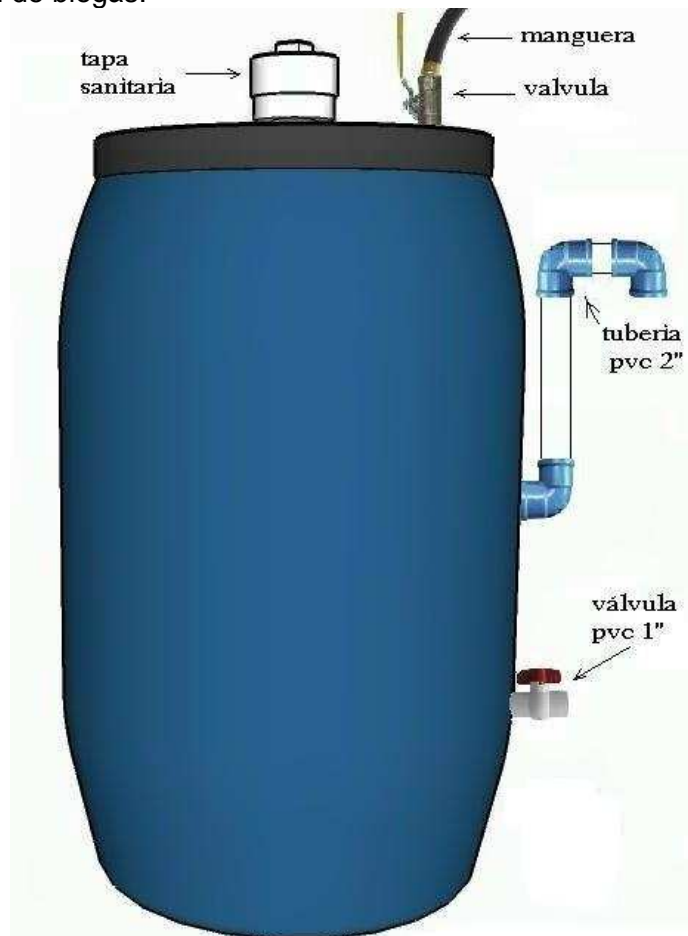


Figura N°9.7 - Biodigestor de carga intermitente tipo Batch casero.

### 9.5.2 - Biodigestor semi-continuo

Es el más usado en áreas rurales, estos biodigestores se cargan diariamente con una carga relativamente pequeña en comparación al total contenido en el biodigestor. Al mismo tiempo se saca de la cámara de descarga un volumen igual de líquido, para mantener el volumen constante.

La producción de biogás es generalmente permanente, debido al constante suministro de nutrientes para las bacterias metanogénicas, responsables de generar el gas. El

único factor limitante sería la disponibilidad de agua, debido a que la carga entrante debe ser en promedio de 1:4 material.

Estos sistemas permiten retirar sustrato ya fermentado por el canal de salida y añadir nueva materia orgánica por el canal de entrada (sin destapar la boca central) que al descomponerse generara nuevo gas. Estos digestores se adaptan bien para tratar material blando, como estiércol de origen pecuario e inclusive humano, materia celulósica, etc.

Ejemplos de estos son el tipo Hindu, figura N°9.8 y el tipo chino figura N°9.9

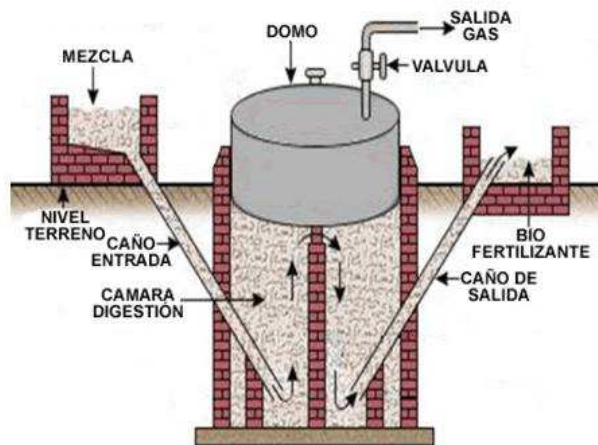


Figura N° 9.8 - Esquema del biodigestor tipo Hindú o de cúpula flotante.

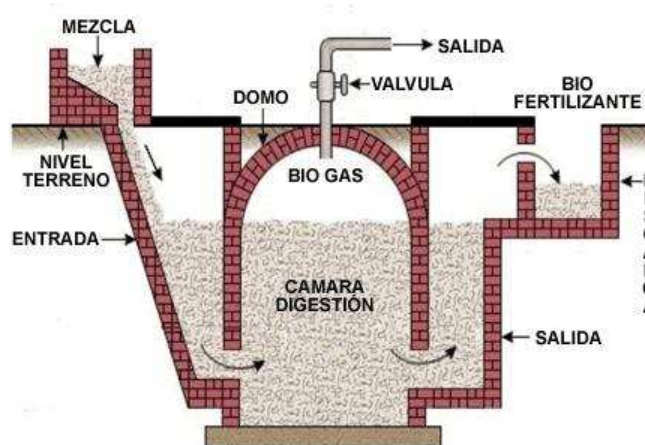


Figura N° 9.9 - Esquema del biodigestor tipo Chino o de cúpula fija.

### 9.5.3 - Biodigestor Continuo

Se mencionara el de desplazamiento Horizontal, ya que biodigestores de régimen continuo como este, el de tanque múltiple y tanque vertical se aplican en industrias, comunas o grandes granjas. Se usa mayor tecnología para el calentamiento del sustrato, control de la calidad del gas resultante, plantas enteras de desulfuración, grandes compresoras, cadenas de distribución, plantas de almacenamiento, antorcha de desfogue y un suministro muy constante de materia orgánica.

### 9.5.3.1 - Desplazamiento Horizontal

Son los que tienen una cámara de digestión alargada, con un perfil cuadrado, rectangular o en V, dependiendo de las características del terreno o conveniencia. En este diseño, los residuos se van degradando a medida que transitan a lo largo del biodigestor. Esta característica lo hace útil en el procesamiento, aprovechamiento de residuos y materiales que requieran un tratamiento prolongado, tales como excretas humanas, desperdicios de sacrificio de animales o material vegetal muy fibroso o difícil de degradar.

El biodigestor Taiwán tubular puede clasificarse como de desplazamiento horizontal por su forma y por el desplazamiento del material, pero esta clasificación es según el régimen de carga, por lo tanto, es semi-continuo.

Se utilizan generalmente para explotaciones agropecuarias que generan grandes cantidades diarias de residuos, como tambos, criaderos de porcinos, granjas de gallinas ponedoras en jaulas, establos de vacunos, etc. También sirven muy bien para pequeñas comunas que desean de estabilizar y neutralizar los residuos orgánicos municipales antes de verterlos al ambiente.

Un biodigestor de desplazamiento horizontal de pequeña escala, se recomienda por cuestiones de diseño, con volúmenes mayores a los quince 15 m<sup>3</sup> aproximadamente. Porque la excavación de un pozo en forma vertical puede resultar problemática, si el nivel de la napa freática está muy cercano a la superficie, puede aparecer demasiada humedad. El agua circula en flujo pistón, similar a la circulación producida en el intestino del ser humano o de algún otro mamífero.

Este tipo de flujo permite que cada porción del residuo que ingresa por un extremo cumpla el tiempo de residencia necesario dentro del biodigestor antes de salir por el otro extremo.

## 9.6 - Estudio de mercado

### Biodigestores Nacionales [71][72][73][74][75]

El presente estudio está centrado en tres marcas de biodigestores que se comercializan en la Argentina y un biodigestor del exterior del país. Las marcas son ROTOPLAS, ETERNIT, EG-INGENIERÍA y GEO.

En la tabla N° 9.4 se presenta un resumen de especificaciones de los biodigestores disponibles en el mercado:

	ROTOPLAS	ETERNIT	EG-INGENIERÍA			GEO
Modelo	BDR 600/1300/3000	Biodigestor de Flujo Continuo	Biodigestor con desplazamiento horizontal	Biodigestor con Geomembranas	Biodigestor EGSB Prefabricado	Biodigestor Geo
Material del reactor	Polietileno	Polietileno	Mampostería	Polietileno de Alta Densidad	Acero inoxidable	Polietileno de Alta Densidad

<b>Rango de temperaturas soportadas</b>	20°C a 60°C	Para ser utilizados a temperatura ambiente	Para ser utilizados a temperatura ambiente	Para ser utilizados a temperatura ambiente	40°C a 70°C	T < 60°C
<b>Rango de presiones soportadas</b>	No lo especifican	No lo especifican	No lo especifican	No lo especifican	No lo especifican	No lo especifican
<b>Capacidad</b>	De 600, 1300 y 3000 litros	850 litros	15000 litros	A partir 750000 litros	300000 litros	A partir de 6000 litros
<b>Material de la tubería</b>	PVC	Polietileno	No lo especifican	Polietileno de Alta Densidad	Acero inoxidable	PVC
<b>PH</b>	No lo especifican	No lo especifican	No lo especifican	No lo especifican	No lo especifican	No lo especifican
<b>Tiempo de retención</b>	Entre 12 y 24 meses	Entre 12 y 24 meses	25 a 40 días	25 a 40 días	25 a 40 días	25 a 30 días
<b>Carga Diaria</b>	Calculado para desechos de 20, 50 y 100 personas	Calculado para desechos de 30 a 40 personas	Carga continua	Carga continua	Carga continua	95 a 100 litros de Desechos
<b>Producción de Biogás</b>	No	No	No lo especifican	No lo especifican	180 m <sup>3</sup>	3 m <sup>3</sup>
<b>Producción de Biol</b>	100, 200 y 800 litros	Si	Si	Si	Si	120 Litros de Abono diluido (Aprox. 10-12 Kg)
<b>Mantenimiento</b>	Cada año	Cada año	No especifican el tiempo	No especifican el tiempo	No especifican el tiempo	Cada año
<b>Vida útil</b>	Más de 35 años	No lo especifican	No lo especifican	No lo especifican	No lo especifican	7 años

**Tabla Nº 9.4 - Resumen de especificaciones de biodigestores disponibles en el mercado.**

Se observa que los biodigestores de Rotoplas y Eternit solo tratan los desechos de aguas residuales, desaprovechando el biogás que se pueda llegar a generar. Los biodigestores de GEO y EG-INGENIERÍA apuntan su producto a zonas agropecuarias, poseen grandes dimensiones y son fabricados según la necesidad de sus clientes. Estos biodigestores no tienen sensores de temperatura, presión y Ph.

Se buscará aplicar la electrónica necesaria para el control del proceso de fermentación con un prototipo de dimensiones reducidas.

## 9.7 - Definición del producto

Dado el segmento de mercado al que se orientará el producto granjas de pequeña y mediana escala, las especificaciones que debería cumplir son las que se muestran en la tabla N° 9.5:

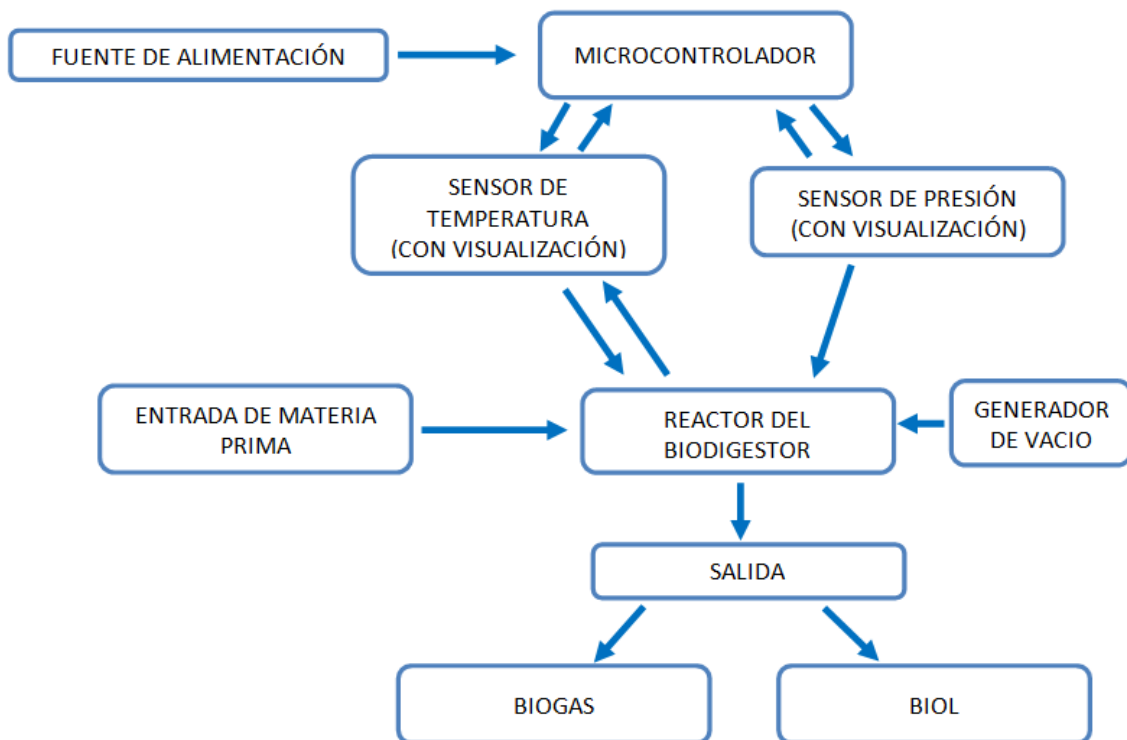
	<b>Características</b>	<b>Escala familiar</b>
<b>Material del reactor</b>	Resistente a la presión y temperatura, liviano, duradero en el tiempo, económico.	Polietileno
	Rango de temperaturas soportadas	20°C a 40°C
	Rango de presiones soportadas	50 KPa
<b>Capacidad</b>	Cámara de digestión + cámara de biogás	200 litros
<b>Material de la tubería</b>		PVC
<b>Sensor de presión</b>	MPX2050DP	SI
<b>Sensor de Temperatura</b>	DS18B20	SI
<b>PH</b>	Valores ideales de trabajo (No controlado)	6.5 a 7.5
<b>Resistencia</b>	Aceleradora del proceso	SI
<b>Tiempo de retención</b>	Con el uso de la resistencia, se obtiene el menor tiempo.	20 a 40 días
<b>Carga</b>	1ra carga (luego cargas diarias de 2.5 litros)	100 litros de desechos
<b>Producción de Biogás</b>	Luego del 1er tiempo de retención	0.075 m <sup>3</sup>
<b>Producción de Biol</b>	Luego del 1er tiempo de retención	3 Litros de Abono diluido (Aprox. 2-3 Kg)
<b>Mantenimiento</b>		No requiere mantenimiento
<b>Vida util</b>		10 años
<b>Alimentación</b>		220 V / 12 V

**Tabla N ° 9.5 - Especificaciones técnicas.**

La mayoría de los biodigestores del mercado, no cuentan con el control de temperatura y el sensado de la presión, que se incluye en el diseño de este proyecto..

## 9.7 - Desarrollo

Debido a su simplicidad y a su bajo costo, se implementó el biodigestor de carga discontinua o intermitente tipo Batch. La figura N°9.10 muestra el diagrama en bloques del sistema electrónico asociado.



**Figura N° 9.10 - Diagrama de bloques del biodigestor.**

El diseño contará con una fuente de alimentación que suministrará los 5V al microcontrolador. El microcontrolador manejará los sensores de temperatura y de presión. El sensor de presión estará en la parte superior externa del reactor, mientras que el sensor de temperatura, estará sumergido en su parte interna.

La temperatura del proceso será la variable que más impacte en el biodigestor, ya que su valor incidirá en el tiempo de producción de biogás.

La resistencia tiene la función de brindar al biodigestor la temperatura correcta para la fermentación que será controlado por el sensor de temperatura junto con el microcontrolador para que se mantenga en las especificaciones requeridas mediante un control tipo ON/OFF. En principio se utilizará la resistencia eléctrica conectada a una línea de alimentación de 220V, pero luego de que se genere el gas combustible, lo ideal sería utilizar el mismo gas metano para que le entregue al digestor las calorías necesarias para que tenga la temperatura justa de funcionamiento.

La presión del proceso será la variable que se utilizará para caracterizar el comportamiento dinámico de la producción de biogás, ya que a medida que se realiza



la biodigestión de los desechos orgánicos, se va acumulando gas dentro de la cavidad vacía que se encuentra en el biodigestor.

### 9.8 - Diseño mecánico

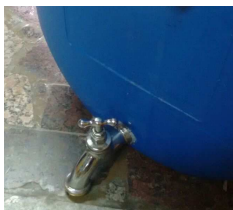
Se realizaron dos prototipos experimentales para el desarrollo del biodigestor teniendo en cuenta los problemas que se presentaron. El prototipo 1 (Figura N° 9.11), al no tener acceso a la parte interior, presentó pérdidas de presión debido a que solo se lo pudo sellar en su parte exterior, lo que no causó una perfecta sujeción de sus partes. En la conexión entre el reactor y el recipiente contenedor de biogás, al realizarlo con un caño muy ancho, no se pudo implementar un mecanismo que permitiera cerrar sus aberturas para separar los recipientes. Debido a estos problemas, se buscó una mejor solución, para ello se usó un recipiente de 200 litros con tapa desmontable ajustable con abrazadera, ver figura N° 9.12.

#### Prototipo 1



Figura N°9.11- Modelo Físico.

- 1) Puesta en funcionamiento: se rellena la cámara de digestión con excremento y residuos orgánicos. Luego se elimina el oxígeno de la cámara de gas utilizando un gas de barrido que lo desplace, el mismo se encontrará en el recipiente pequeño, que se puede retirar después de la puesta en marcha.
- 2) Borne de la resistencia
- 3) Sensor de temperatura



**Figura N°9.13 - Salida de BIOL.**



**Figura N°9.14 - Salida de BIOGAS**

Las pérdidas de presión se presentaron en todas las partes mecánicas añadidas del recipiente. Hubo pérdidas de presión en la resistencia, en la canilla de salida de biol (Figura N°9.13) y la llave de salida de biogás (Figura N° 9.14). Se produjeron porque resultó difícil el sellado de todas las partes mecánicas. Por ello, se optó por una segunda alternativa.

### **Prototipo 2**

- 1) Puesta en funcionamiento: se debe cerrar la tapa con la abrazadera, ver figura N°9.15. En el medio del recipiente (Figura N°9.16), se puede observar la manguera del sensor de presión y en su parte inferior, el sensor de temperatura y la resistencia.
- 2) Se desenrosca la tapa para poder ingresar la materia prima, (Figura N°9.11). Se elimina el oxígeno de la cámara de gas con el recipiente externo que posee dióxido de carbono.
- 3) Una vez eliminado el oxígeno, se retira la manguera del recipiente externo y se cierra la etapa de entrada de materia prima para que pueda empezar el proceso de fermentación.
- 4) Pasado el tiempo de fermentación, se puede utilizar el biogás generado en la hornalla y retirar el biol por su respectiva salida, figura N°9.16 y figura N°9.17.



**Figura N° 9.12 - Recipiente de 200 litros.**



**Figura N° 9.15 - Abrazadera que ajusta la tapa.**



**Figura N°9.16 - Resistencia, Sensores de Temperatura y Presión.**



**Figura N° 9.17 -Salida de BIOL.**



**Figura N°9.18 - Entrada de materia prima y BIOGAS**



**Figura N° 9.19 - Hornalla para el salida del biogás.**

Para el almacenamiento del gas, en el Prototipo 1 se utilizó un recipiente de iguales dimensiones que el reactor. De esta manera, todo el biogás producido se envía por un caño de plástico hacia el otro recipiente de almacenaje de biogás. En el prototipo 2, el gas se encuentra en la parte superior del mismo recipiente. Como los prototipos utilizados son de flujo de carga discontinua y de estructura rígida, el gas no se puede expandir y por lo tanto la presión irá aumentando a medida que se produce el gas, se puede así obtener un análisis del volumen de producción de biogás a partir del aumento de presión en el interior del biodigestor.

El reactor contará una entrada de desechos y con salidas para el gas metano y el fertilizante. La salida para el biogás estará en la parte superior con una caño de gas de acero que estará conectado directamente con el sensor de presión, mientras que la salida del biol, será en la parte inferior por medio de una válvula de plástico, para que se desagote fácilmente cuando sea necesario.

Para realizar el vacío de aire del reactor, se puede utilizar una mezcla de agua, levadura y azúcar que genera dióxido de carbono. Se puede realizar con un recipiente externo al biodigestor y se lo puede conectar con una manguera de plástico en la parte superior del biodigestor, este método es el que se utilizará en el proyecto. Otra

alternativa de mezcla, sería una máquina o motor que produzca CO<sub>2</sub>. Se debe realizar sólo antes de efectuar la primera carga.

### 9.9 - Microcontrolador

Para el desarrollo del biodigestor no se requieren altas velocidades de procesamiento, pero si es importante el número de entradas/salidas y el tamaño de la memoria de programa debido a las librerías correspondientes a los sensores.

Para este trabajo se necesita un microcontrolador que posea al menos 6 entradas/salidas analógicas y una memoria flash mayor a 16 Kb para el almacenamiento del programa y futuras actualizaciones del proyecto. Se ha optado por el Arduino Uno el cual posee un CPU de 16 Mhz, 6 entradas/salidas analógicas, una memoria flash de 32 Kb y una rápida y sencilla programación en comparación con otros microcontroladores del mercado (Figura N° 9.20).

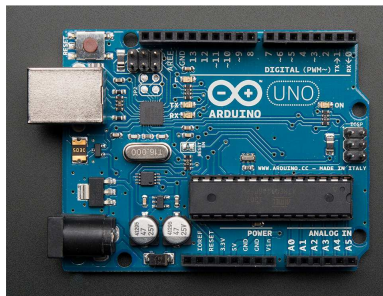


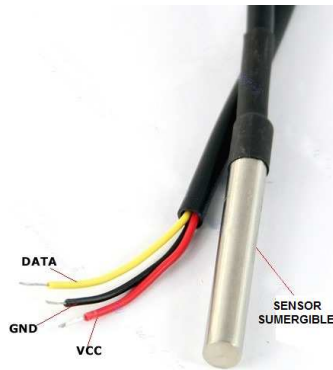
Figura N° 9.20 - Arduino UNO.

### 9.10 - Sensor de Temperatura

Para el sensado de la temperatura se estima que será necesario cumplir con los siguientes requisitos:

- Sumergible
- Salida digital
- Lectura lineal de la temperatura
- Resolución en grados Celsius mayor a 8 bits
- Precisión de aproximadamente  $\pm 5$  °C o superior
- Rango de temperatura de trabajo 20 °C a 40 °C.
- Tensión de alimentación 3V o 5.5 V.

Se seleccionó el sensor digital con cable sumergible en fluidos DS18B20 (Figura N°9.21). Su salida es digital, la lectura de la temperatura es lineal, su resolución es de 9 a 12 bits con una precisión de  $\pm 5$  °C. Trabaja en un rango de temperatura de -10 °C a +85 °C y puede trabajar bajo la misma tensión de alimentación del microcontrolador.



**Figura N° 9.21 - Cables para la conexión y sensor sumergible.**

Cuenta con un cable por el que serán recibidos los datos sensados, un cable GND, un cable de alimentación y el sensor sumergible (Figura N° 9.21).

### 9.11 - Sensor de Presión

Para el sensado de la presión, se necesita un sensor que sea capaz de realizar la medición dentro del biodigestor, ya sea mediante una sonda o manguera. Debe soportar una presión de 10 KPa. Debe ser analógico, para que se pueda tomar una diferencia de potencial, leerla por unas de las entradas del microcontrolador y realizar la correspondiente tabla de valores para el posterior muestreo de los datos en los displays 7 segmentos.

Estas características las cumple el sensor analógico diferencial MPX2050DP(Figura N°9.22).



**Figura N° 9.22 - Sensor de presión diferencial MPX2050DP.**

Este toma dos medidas. La primera, mide la presión ambiente. La segunda, mide la presión del recipiente mediante una pequeña manguera que se coloca en el interior del biodigestor, luego realiza internamente la diferencia entre ambas medidas. De esta manera se obtiene una diferencia de potencial que se puede aplicar a unas de las entradas del microcontrolador y obtener el valor de presión medido. Tiene una sensibilidad de 0.8 mV / KPa. Puede llegar a medir presiones de hasta 50 KPa.

### 9.12 - Resistencia

Para poder acelerar el proceso de fermentación y creación de gas metano, se recurre a una resistencia eléctrica sumergible en fluidos de 1800 W / 220 V(Figura N° 9.23). Se busca mantener una temperatura dentro del reactor del biodigestor que ronde entre 25 y 40 °C. Será controlada por el microcontrolador, cuando el reactor este por debajo de 25 °C, junto con un relé de 5V, la resistencia se pondrá en funcionamiento. Cuando

el reactor tenga una temperatura de 40 °C, la resistencia se desconectará automáticamente.



Figura N° 9.23 - Resistencia.

### 9.13 - Esquemáticos, PCB y distribución de los componentes

Se representan los esquemáticos (Figura N° 9.24), pcb(Figura N° 9.25 y 9.28) y la distribución de los componentes de los displays y los sensores (Figura N° 9.26 , 9.27 y 9.29).

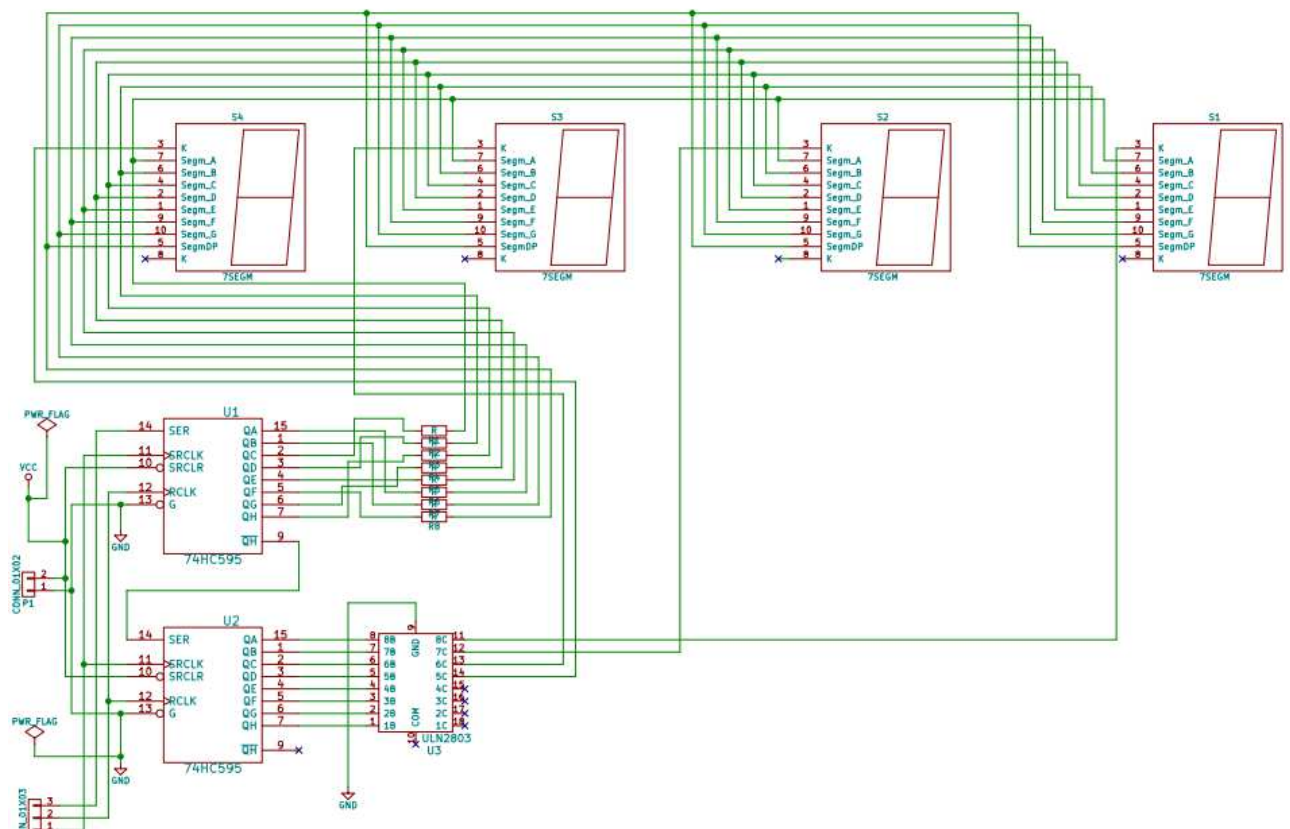


Figura N° 9.24 - Esquemático de los displays

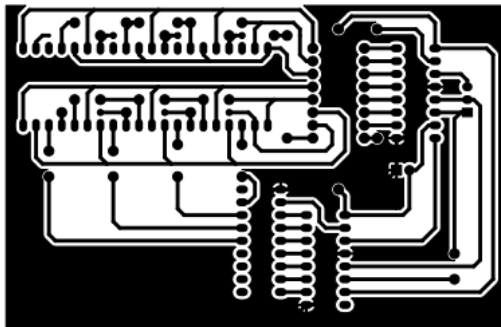


Figura N°9.25 - PCB de los displays.  
de los

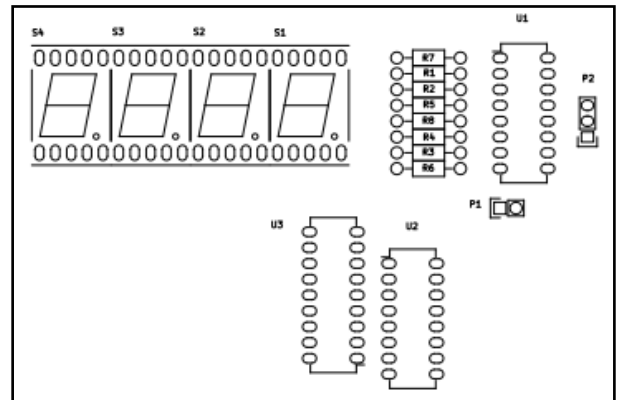


Figura N° 9.26 - Distribución  
componentes de los displays

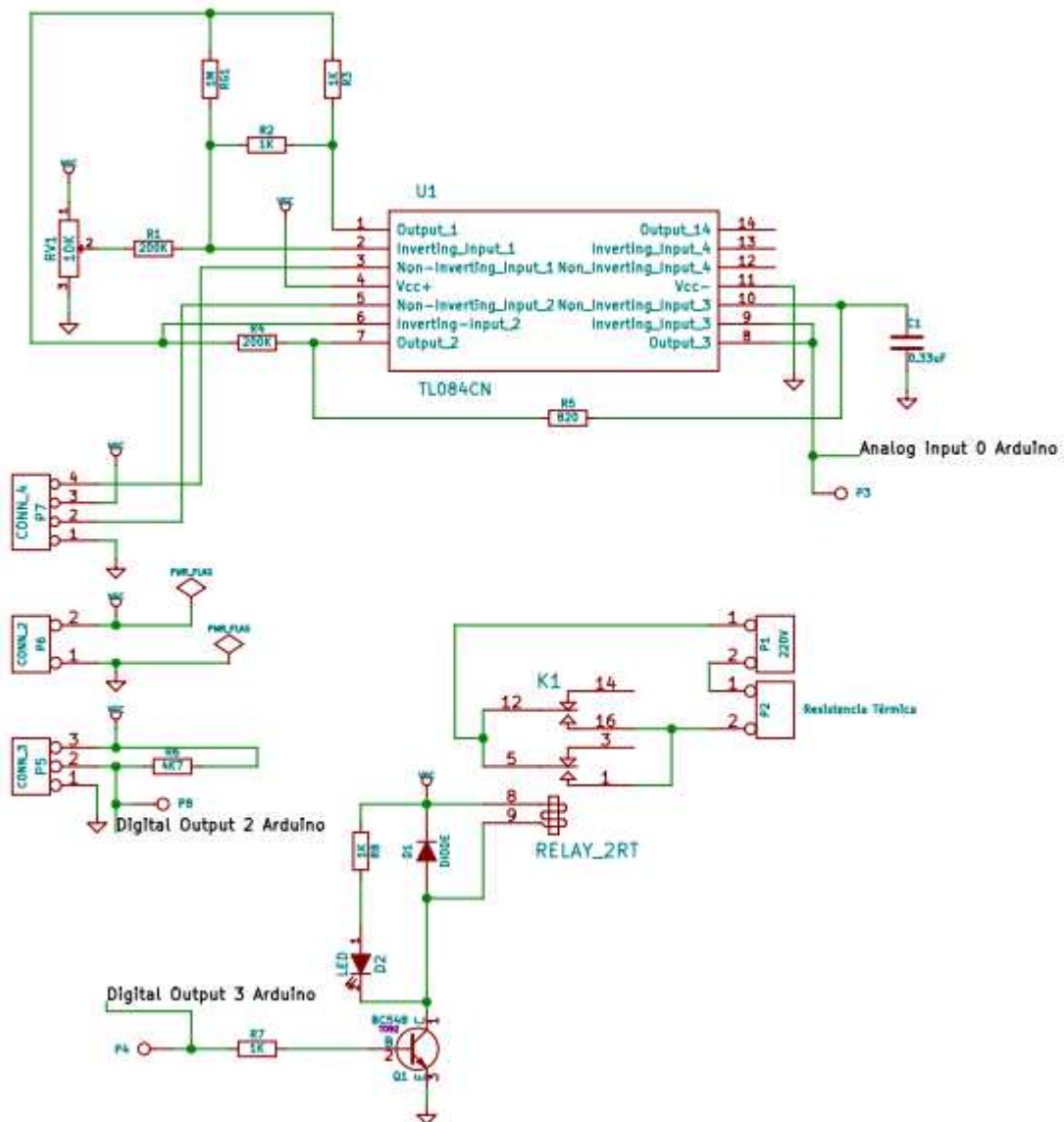


Figura N° 9.27 - Esquemático de los sensores.

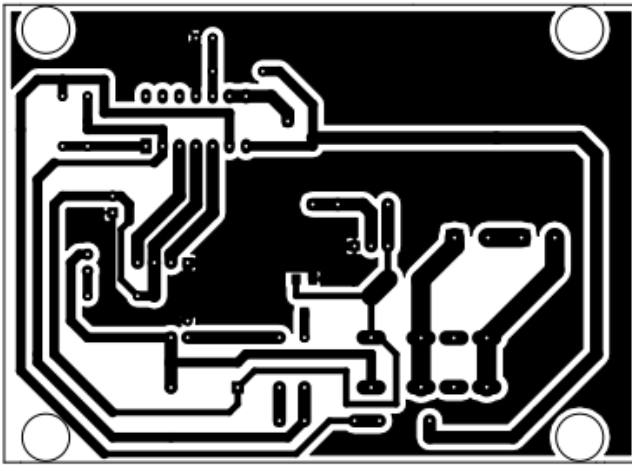


Figura N°9.28 - PCB de los sensores.

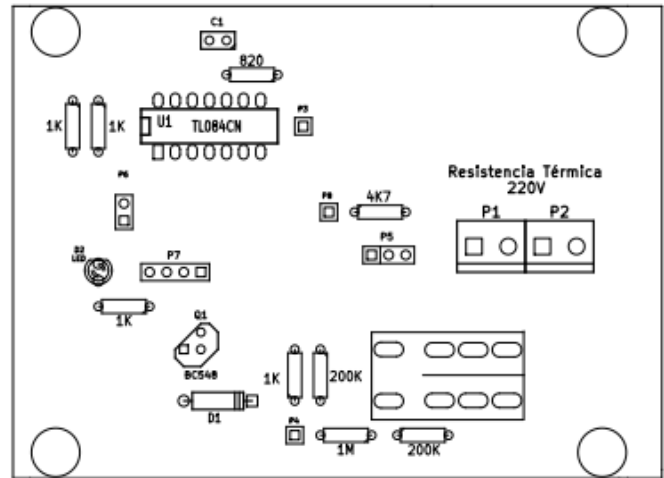


Figura N° 9.29 - Distribución de componentes del sensor.

La Figura N° 9.30 nos presenta el sistema terminado

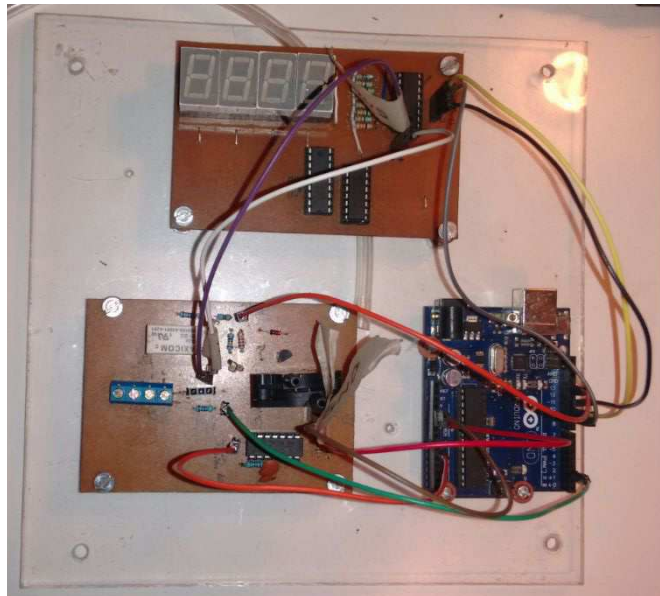


Figura N° 9.30 - Placa vista desde su parte superior.

### 9.14 - Firmware

La figura N° 9.31, presenta un diagrama de flujo del firmware del microcontrolador. Este chequea constantemente si el pulsador fue presionado. En ese caso, comienza a actuar un software anti rebote y si en ese momento los displays mostraban los valores medidos de presión, pasa a mostrar automáticamente los valores de temperatura y viceversa.



Una vez que se chequea si hay un pulso provocado por los pulsadores, se sensa la presión, se calcula su valor y se lo guarda en un buffer para su posterior actualización en los displays. Luego se sensa el valor de temperatura, se calcula su valor y se lo guarda en un buffer para su posterior actualización en los displays.

Sensados los valores de presión y temperatura de ese instante, el microcontrolador chequea en que nivel de temperatura se encuentra el reactor del biodigestor. Si se encuentra comprendido entre 25°C y 35°C la resistencia permanece encendida. En caso contrario, la resistencia se apaga.

Luego se refrescan y se actualizan los valores de los displays con los parámetros actualizados de presión y temperatura. Por último, se realiza la transmisión UART (Universal Asynchronous Receiver-Transmitter) a USB (Universal Serial Bus) hacia la PC. Esta transmisión envía los valores medidos de presión y temperatura, y el ciclo comienza nuevamente.

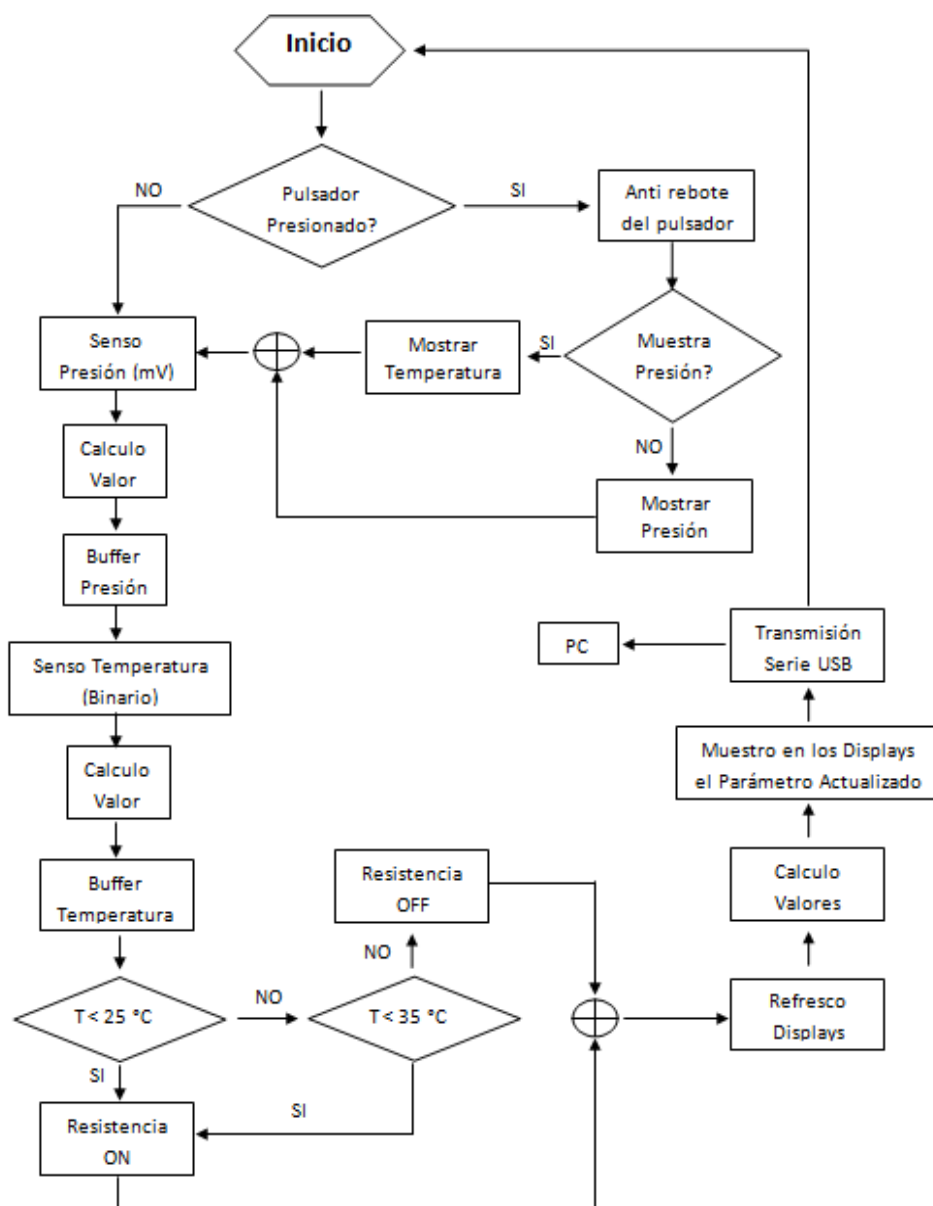


Figura N°9.31 - Diagrama de flujo del firmware.

### 9.15 - Visualización

Se ha hecho un pequeño programa con Matlab 2014a que consiste en la lectura de los datos de presión en KPa y temperatura en °C que envía Arduino por el puerto serie, y ofrece una visualización gráfica de dichos datos en función del tiempo.

El software es sencillo, una vez ejecutado se configura el puerto serie y si los parámetros son correctos se abre el puerto. Luego comienza un bucle iterativo infinito en el cual se hace la lectura de los datos y se los acomoda en dos gráficos para la visualización en tiempo real, además existe con la posibilidad de guardar dichos gráficos utilizando las herramientas otorgadas por Matlab. Una vez que se llega a una cierta cantidad de valores los gráficos vuelven a tiempo cero.

En la figura N°9.32 se ilustra el diagrama de flujo del software.

En la figura N°9.33 se muestra una captura de pantalla con las gráficas de presión y temperatura.

### 9.16 - Ensayo de los Sensores

Antes de detallar los ensayos realizados se presenta un estudio que trata sobre la producción de biogás y biol sin realizar control de temperatura. Fue realizado por la Universidad Politécnica de Madrid y la Universidad Técnica Particular de Loja [76].

Se implementó un prototipo de flujo de pistón (Figura N° 9.34). Unos de los ensayos que se propuso fue producir biogás según las necesidades energéticas a abastecer. Se tomó en cuenta a una familia que necesitaba combustible para cocinar durante cuatro horas cada día y que contaba con una explotación de cinco vacas. El biogás necesario para abastecer a esta familia era de 520 litros de biogás diario. Y la carga estimada diaria para la producción de biogás era de 15 Kg/día.

Para la carga del Biodigestor se mezclaron esos 15 Kg con tres partes de agua, 45 litros de agua diarios. Se consideró la temperatura ambiente de 20°C y unos 40 días de retención, se obtuvieron los resultados que se indican en la figura N° 9.35.

En la parte introductoria se mencionó que todos los biodigestores investigados, necesitan realizar un control de temperatura que mantenga al reactor entre los 20°C y los 35°C. Esto permite acelerar el proceso de biodigestión, se obtiene mayor producción de biogás, mejor calidad de biol y menores tiempos de retención. Para independizar la producción de biogás del factor climático, en éste proyecto se dispuso de un método para mantener la temperatura en un rango óptimo de trabajo, dicho rango de temperatura se encuentra entre los 25°C y los 40°C (proceso mesofílico). El método empleado consiste en activar una resistencia cuando la temperatura disminuye de un valor predefinido en 25°C, luego se desconecta cuando la temperatura empieza a subir o se encuentra por encima del valor estipulado.

En la figura N° 9.36, se puede observar los datos obtenidos en la producción de biogás en el lapso de tiempo de 30 días. Se puede apreciar el máximo valor de presión obtenido de 9 KPa en el día 20.

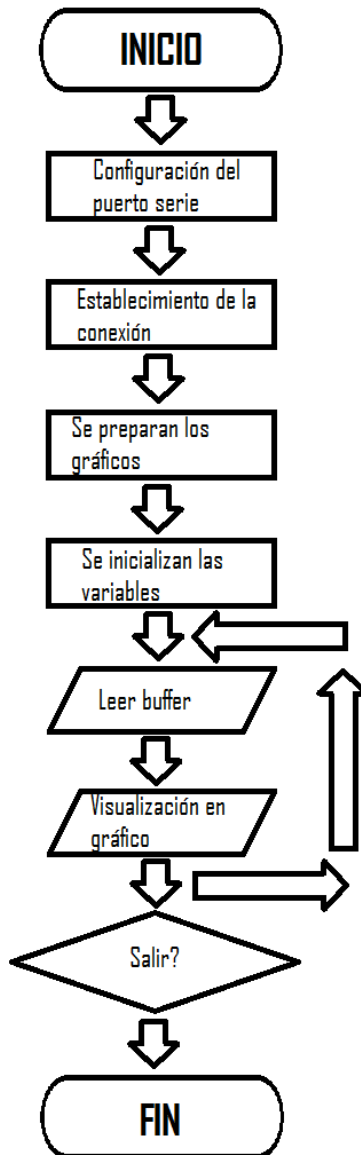


Figura Nº 9.32 - Diagrama de flujo del Software.

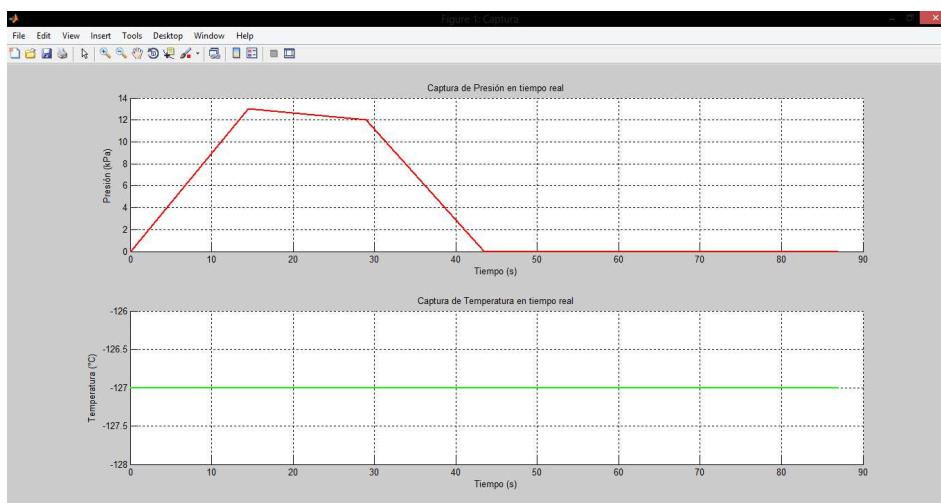
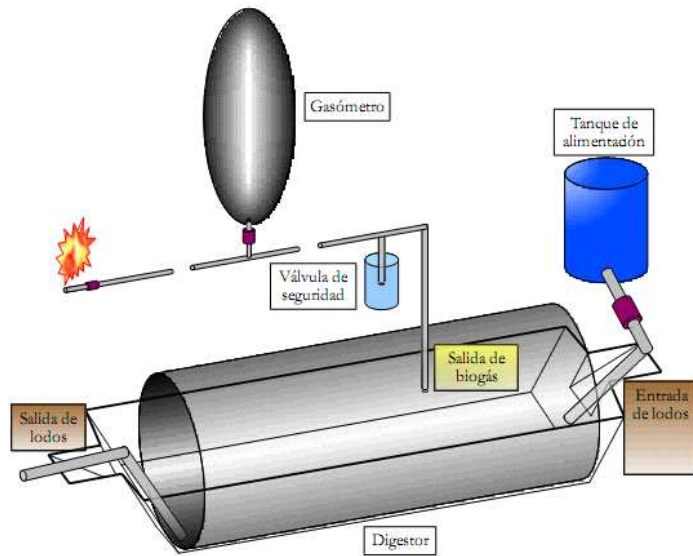


Figura Nº 9.33 - Gráficas de presión y temperatura.



**Figura Nº 9.34 - Biodigestor de flujo de pistón.**

Carga diaria: 15 kg de estiércol de vaca diluidos en 45 litros de agua.	
Temperatura de trabajo: 20°C.	
Tiempo de retención (con mejora de fertilizante): 40 días.	
Producción diaria de biogás: 520 L.	
Producción diaria de biol: 60 L.	
Volumen líquido:	2,4 m <sup>3</sup>
Volumen gaseoso:	0,8 m <sup>3</sup>
Volumen total:	3,2 m <sup>3</sup>
Ancho de rollo:	1,25 m
Longitud del digestor y de la zanja:	6,40 m
Relación L/D:	8,00
Ancho inferior de la zanja:	0,70 m
Ancho superior de la zanja:	0,90 m
Profundidad de la zanja:	0,70 m
Volumen del reservorio de gas:	1 m <sup>3</sup>
Cantidad total de plástico a comprar (incluido el reservorio):	15 metros de rollo de 1,25 m de manga, y 4 metros de manga de 1,5 m.

**Figura Nº 9.35 - Resultados del biodigestor propuesto.**

El total de biogás producido se puede calcular de la siguiente manera:

Volumen de gas dentro del biodigestor:

$$V = 0.25 \times 200 \text{ l} = 50 \text{ l} \quad (15)$$

Presión de gas metano (67 % del gas producido)

$$P_{\text{parcial}_{\text{CH}_4}} = 0.687 \times 9.2 \text{ KPa} = 6.2376 \text{ KPa} = 0.06156 \text{ Atm} \quad (16)$$

A una temperatura de 35°C (308 °K)

$$n = \frac{PV}{RT} = \frac{0.06156 \times 50}{0.082 \times 308} = 0.1219 \text{ mol de } CH_4 \text{ (17)}$$

$$\text{Gramos de } CH_4 = \frac{0.1219 \text{ mol } CH_4 \times 16 \text{ g}}{1 \text{ mol } CH_4} \text{ (18)}$$

La densidad CH<sub>4</sub> es 0.66 Kg/m<sup>3</sup>, se obtiene la cantidad de metano generado

$$\text{Metano (m}^3\text{)} = \frac{2 \text{ g} \times 1 \text{ m}^3}{660 \text{ g}} = 0.003 \text{ m}^3 \text{ (19)}$$

Debido a que solo se realizó una carga, se obtuvo una cantidad de biogás que ronda los 0.003 m<sup>3</sup>.

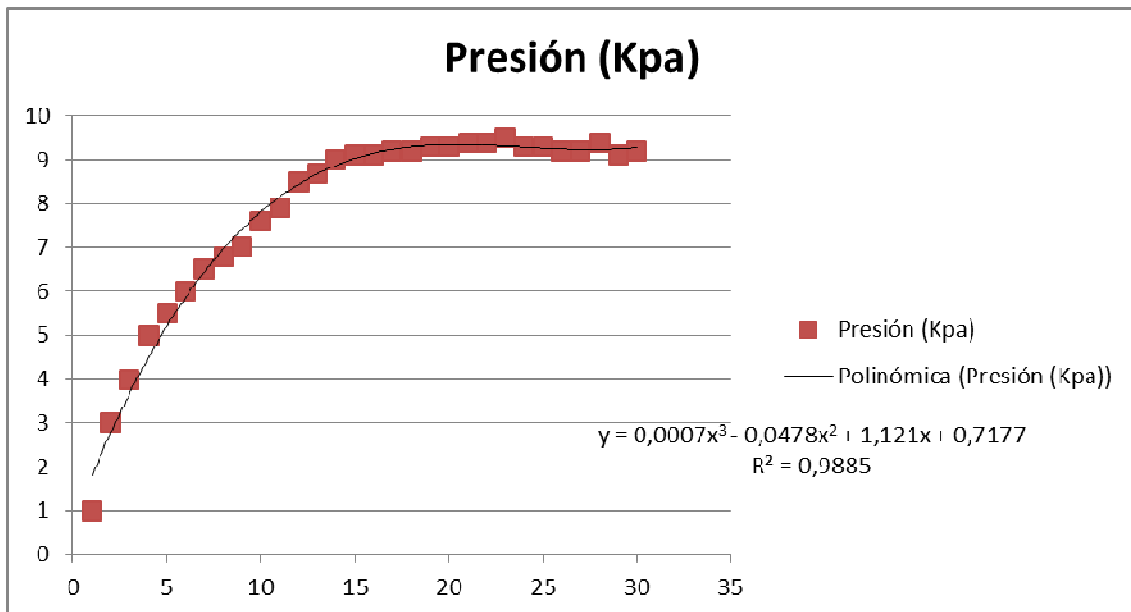


Figura Nº 9.36 - Gráfica que indica el nivel de biogás producido.

### 9.17 Conclusiones

Se ha manifestado a través de este proyecto que el proceso de generación de biogás a temperatura ambiente lleva mucho tiempo, debido a que las condiciones climáticas afectan al proceso haciéndolo más lento (en caso de días frescos) o acelerándolo (en caso de días calurosos). En caso de que el biodigestor esté sometido a temperaturas inferiores de los 20°C el tiempo de retención es alrededor de 2 meses, dependiendo de qué materia prima se utilice para el llenado.

Con el proceso realizado en el proyecto la generación de gas necesita menos tiempo de retención cuando la temperatura empieza a subir. Se expone en este informe que lo investigado y ensayado es certero, ya que se necesitó solo alrededor de 12 días para poder generar gas, siendo este un tiempo relativamente corto comparado con el proceso natural.

En otras investigaciones, se comprobó que el proceso de biodigestión no se puede llevar a cabo a elevadas temperaturas ya que se quemarían las bacterias de la materia prima que llevan a cabo el proceso de generación de biogás, perdiendo validez y eficiencia.

Se demuestra en este proyecto que el proceso no es permanente, es decir que hay que recargarlo para que el proceso tenga fluidez. En el proyecto se cargó una sola vez, lográndose una presión máxima de 9KPa, dicha presión disminuyó con el tiempo debido a una fuga de gas.

Estimando cargas diarias de 2.5 litros de desechos, la producción de biogás llegará a los valores deseados de 0.075 m<sup>3</sup> de biogás, lo que significa que tendrá una equivalencia energética de 0.165 KW/h. La resistencia eléctrica para calentar el reactor del biodigestor tiene un consumo de 0.18 KW/h, que solo se estuvo activando dos veces por día, aproximadamente 15 minutos en el lapso de tiempo de 12 días de la biodigestión.

### **9.18 - Mejoras a futuro**

El diseño actual del biodigestor no controla el nivel de Ph de la mezcla, el cual afecta directamente a la vida de las bacterias encargadas de realizar la digestión del material, realizando cargas diarias de materia prima (y al mismo tiempo extracciones de biol), se asegura mantener el PH en el nivel adecuado. Como mejora se propone incorporar un sistema de control de PH, como indicador del correcto funcionamiento del biodigestor. El biodigestor no cuenta con una paleta o algún sistema mecánico para revolver periódicamente los desechos de su interior. Lo ideal sería contar con un sistema de motor y paleta.

Las reducidas dimensiones del biodigestor implementado en comparación de otros equipos investigados, son un impedimento para la producción de una mayor cantidad de biogás y biol. Si en futuros prototipos se llega a implementar un biodigestor de mayores dimensiones el rendimiento mejorará notablemente.

Una buena mejora sería implementar la carga automática de materia prima, pero para eso se debería contar con un biodigestor y un gasómetro de grandes dimensiones.

## **10- Generador Eólico de baja potencia**

### **10.1- Introducción**

Tal como se ha venido mencionando en la introducción y en apartados anteriores en que se presentaron los trabajos sobre otras tecnologías renovables, durante la última década han surgido o se han acelerado diferentes crisis concurrentes de diversa índole: la crisis del clima, de la diversidad biológica, del combustible, alimentaria, del agua y, finalmente, del sistema financiero y del conjunto de la economía. La escalada de las emisiones que cambian el clima sugiere que la amenaza del cambio climático podría descontrolarse, con posibles consecuencias desastrosas para el hombre.

Una alternativa para afrontar estas crisis son los aerogeneradores, quienes entregan energía limpia y poco contaminante en comparación con los demás procedimientos de obtención de energías. Por ello, el impacto medio ambiental es mínimo. Es importante aclarar que los aerogeneradores durante su vida útil no generan ningún tipo de contaminación, dado que el único momento en el que lo hacen es durante su proceso de fabricación. Por eso se consideran equipos “poco contaminantes”.

Ventajas de los Generadores Eólicos:

- No tiene riesgos respecto de la volatilidad del precio del combustible, ya que su costo es nulo.
- Los gastos de operación y mantenimiento son sumamente bajos.
- No produce gases contaminantes.
- No implica riesgos geopolíticos respecto de restricciones sobre otros países.
- Es inagotable.

Desventajas de los Generadores Eólicos:

- Se encuentra dispersa.
- Es intermitente y aleatoria (no continua)
- Requiere un costo elevado de inversión inicial
- Su mal dimensionamiento en cuanto a potencia puede causar una pérdida de dinero si se consume mucha menos energía para la cual fue diseñado el aerogenerador.

Dado que la generación de energía eléctrica a partir del viento es un método ecológico, eficiente y renovable, y por ende muy ventajoso con respecto a los métodos tradicionales, vemos un campo de aplicación muy amplio alrededor del mundo.

### **10.2- Energía eólica en argentina**

Nuestro país, pionero en Latinoamérica, dispone actualmente de 13 parques eólicos localizados en 6 provincias que suman una potencia instalada de 29,7 MW, registrando un interesante factor de planta medio cercano al 30%, aun cuando los parques funcionando en la Patagonia alcanzan regularmente factores de utilización muy superiores, con valores tan altos como 40% o más. Varios de estos emprendimientos han crecido al amparo de los beneficios fiscales concedidos por el “Régimen Nacional de la Energía Eólica y Solar” introducido por la Ley 25.019/98. El más representativo es seguramente el parque eólico “Antonio Morán” de la Sociedad Cooperativa Popular de Comodoro Rivadavia, que con 24 aerogeneradores en servicio, es uno de los más grandes de Sudamérica. Las perspectivas del país en materia de energía eólica son francamente alentadoras. Se estima que el potencial

eólico patagónico al sur del paralelo 42 encierra una energía decenas de veces mayor al contenido en toda la producción anual argentina de petróleo. Más aún, no solo el extremo sur argentino posee condiciones favorables para la instalación de granjas eólicas, existen asimismo numerosas regiones aptas en las provincias de Río Negro y Neuquén, en varias zonas serranas y costeras de la provincia de Buenos Aires, y en muchos otros sitios puntuales de todo el país [106].

La tabla 10.1 resumen los parques eólicos más importantes de la Argentina en estado operativo, que inyectan la energía generada a la red nacional de distribución, en general, administrados en el MEM (mercado eléctrico mayorista) [107].

Nombre del Parque	Ubicación	Generación Anual Promedio	Potencia	Puesta en Servicio	Aerogeneradores	Propietario
Arauco	Valle de La Puerta, La Rioja	46 GWh	25,2 MW	ene-11	12 x Impsa IWP-83 -2,1 MW	Gobierno de La Rioja (75%) y Enarsa (25%)
			25,2 MW	feb-14	12 x Impsa IWP-83 -2,1 MW	
Diadema	Diadema Argentina, Chubut	28 GWh	6,3 MW	jul-11	7 x ENERCON E-44- 900 KW	Hychico S.A.
Loma Blanca	Trelew, Chubut	183 GWh	51 MW	jul-13	17 x Alstom ECO100de 3 MW	Isolux Corsán S. A.
Necochea EOS	Necochea, Buenos Aires		250 KW	dic-09	1 x MICON - 250 KW	Sea Energy S.A.
El Tordillo	Comodoro Rivadavia, Chubut		3MW	jul-13	1 x NRG 1500 - 1,5 MW	Vientos de la Patagonia I S. A
					1 x IMPSA IWP-70 - 1,5 MW	
Rawson	Rawson, Chubut	296 GWh [1]	48,6 MW	sep-11	27 x Vestas V90 - 1,8 MW	GENNEIA (ex Emgasud)
			28,8 MW	ene-12	16 x Vestas V90 - 1,8 MW	

**Tabla 10.1. Parques Eólicos más importantes de la Argentina en estado operativo.**

Alrededor del mundo, podemos encontrar una gran diversidad de generadores eólicos, los cuales son diseñados para aprovechar las características específicas de cada región.

En nuestro país, este tipo de energía renovable tiene un alto grado de aplicación, debido en parte a las características demográficas de la población, y al gran flujo de viento que recorre gran parte de nuestro territorio. De las distintas posibilidades de disponibilidad de esta energía generada, mencionaremos tres casos que, al menos en nuestro país, conforman situaciones generalizadas y bien caracterizadas:

- Vastas zonas del país tienen población y establecimientos rurales y de servicios dispersos tales como estancias, escuelas, puestos de gendarmería, policías dispensarios médicos, etc. Como la electrificación rural, muchas veces no rentable para las empresas de distribución de energía eléctrica, normalmente esta gente y establecimientos no tienen acceso a una provisión energética segura y confiable. Este recurso posee una gran importancia ya que



permite la integración de los sectores rurales al desarrollo económico nacional, frenando la migración rural-urbana y mejorando el nivel socio cultural de los habitantes. Mediante máquinas de pequeña potencia (generalmente menos de 10 kW) se puede proveer, si la calidad del recurso eólico es adecuada, de energía para mejorar las condiciones de vida posibilitando el acceso a iluminación, comunicación social (televisión, radio, etc.) y de seguridad y eventualmente la utilización de algunas pequeñas herramientas eléctricas. Desde este punto de vista, es sencillo observar la gran oportunidad que este tipo de energía representa para las vastas zonas rurales que se extienden a lo largo de nuestro país.

- La existencia de un importante número de pequeñas localidades donde ya se cuenta con un servicio eléctrico de origen térmico, en general accionado a gas oíl, el que en muchos casos por razones de costo y protección del equipo se utiliza solamente unas pocas horas al día, impidiendo de esta manera el eventual establecimiento de pequeñas industrias derivadas de la actividad local y retrasando también las posibilidades de mejor condición de vida de la población involucrada. Si las circunstancias lo permiten, acoplado máquina eólicas de una capacidad acorde con la de los equipos existentes, se puede mejorar la calidad del servicio aumentando su confiabilidad y prestación, con el agregado de poder lograr una disminución en el consumo de combustible.
- En regiones con recurso eólico adecuado, si se cuenta con una red de transmisión de alta tensión, es posible establecer verdaderas “centrales” eléctricas conformadas por un número variable, pero en general importante, de turbinas de elevada potencia (75 kW en adelante). La energía generada en estas instalaciones, denominadas granjas o parques eólicos, ingresa directamente a la red.

### **10.3- Generadores eólicos**

Un aerogenerador es un generador eléctrico movido por una turbina eólica, donde la energía cinética del aire en movimiento, proporciona energía mecánica a un rotor hélice que, a través de un sistema de transmisión mecánico, además convierte la energía mecánica rotacional en energía eléctrica. Para aportar esta energía a la red eléctrica, los aerogeneradores deben estar dotados de un sistema de sincronización para que la frecuencia de la corriente generada se mantenga perfectamente sincronizada con la frecuencia de la red. Este último sistema no fue abordado en este trabajo.

Existen distintos tipos de aerogeneradores, y pueden trabajar de manera aislada o agrupados en parques o plantas de generación eólica.

En los aerogeneradores de eje horizontal, el eje de rotación del equipo se encuentra paralelo al piso y además, su eje de rotación principal se encuentra en la parte superior de la torre, orientándose, en el caso de los aerogeneradores pequeños mediante una veleta, mientras que los más grandes utilizan un sensor de dirección y se orientan por servomotores o motorreductores.

En los aerogeneradores de eje vertical, su eje de rotación se encuentra perpendicular al suelo, lo cual provoca que ocupen menos superficie, de modo que se pueden situar más cerca unos de otros. Unas de sus ventajas son sus palas omnidireccionales debido a las cuales no se necesita ningún mecanismo de orientación, además de que pueden colocar más cerca del suelo ya que son capaces de funcionar con una menor velocidad del viento.

#### 10.4- Dimensionamiento del aerogenerador.

Una de las tareas más importantes a la hora del diseño de un generador eólico el dimensionamiento del mismo tomando como referencia la potencia que se necesitará en el espacio físico donde se lo instale. Una elección errónea del generador podría generar la destrucción del mismo. Por ejemplo, si se diseñara un generador eólico de potencia nominal de 1000W (el cual suministra 67 KW/h a una velocidad media del viento de 3 m/s) y se lo ubicara en un sitio donde el consumo sea de aproximadamente 30 KW/h, el generador estaría entregando siempre más potencia de la utilizada, por lo tanto la corriente generada se estaría derivando constantemente a una resistencia de descarga la mayor parte del tiempo desgastando de forma apresurada el sistema de regulación y almacenamiento, o causando incluso, daños al propio generador.

En la investigación acorde a qué generador realizar nos encontramos con dos opciones: de potencia nominal de 350W y de potencia nominal de 700W. Cada uno de ellos con un costo aproximado de \$7500 y \$9500 respectivamente. Para determinar cuál sería más conveniente realizar se hizo un estudio de consumo como se detalla en la siguiente tabla N°10.2 [108].

Consumo promedio diario	Potencia (W)	Energía consumida en 1 hora (W/h)	Horas / día	Energía consumida en un día (W/h)
Heladera mediana	150	150	24	3600
15 Luces de bajo consumo (15W c/u)	225	225	6	1350
Televisor color 20"	70	70	4,5	315
Ventilador de techo	60	60	6	360
Reproductor de video	100	100	7	700
Minicomponente	60	60	1	60
<b>Total</b>	<b>665</b>	<b>665</b>		<b>6385</b>

**Tabla 10.2. Estudio sobre el consumo energético de un hogar promedio.**

Se puede observar que si se realizara un generador eólico de potencia nominal de 350W solo se podría alimentar muy pocos artefactos, y solamente esto pasaría en las mejores condiciones climáticas donde haya un buen recurso eólico, por lo tanto se decidió realizar el generador eólico de 700W, ya que además de su buen aporte de energía, no existen grandes diferencias de costos con respecto al de 350W.

#### 10.5- Estudio de mercado

En el mercado actual podemos encontrar diferentes modelos de Generadores Eólicos, entre los cuales encontramos los modelos Hugo Piggott, ST Charger, el grupo AVL, el IVS 4500 de Invap Ingeniería o el modelo Eolux de la empresa Giacobone entre otros.

En la tabla N°10.3 detallaremos algunas marcas de aerogeneradores reconocidas en el mercado, sus modelos y características principales.

ESPECIFICACIONES	MODELO		
	ALEKO® WG700	NE- 600	Ista-Breeze
Potencia Nominal	700 W	600 W	700 W
Número de Hélices	2	3	3
Diámetro Hélice	1,7 metros	SIN ESPECIFICAR	1,5 metros
Material Hélice	SIN ESPECIFICAR	Fibra de Nylon	Nylon y Fibra de Vidrio
Voltaje de la Batería	12 V/24 V	12 V	24 V
Velocidad del viento nominal	10 m/s	SIN ESPECIFICAR	SIN ESPECIFICAR
Sistema de control	SIN ESPECIFICAR	Electroimán	SIN ESPECIFICAR
Alternador	SIN ESPECIFICAR	Trifásico de imanes permanentes	SIN ESPECIFICAR
Peso	19,5 Kg	15,2 Kg	18 Kg
Precio	US \$449,00	\$ 12.200,00	599,00 EUR

Tabla 10.3. Especificaciones y Precio final de algunos Aerogeneradores que se encuentran en el mercado [109][110][111].

#### 10.5.1- Definición de la potencia del aerogenerador.

Se decidió el diseño de un aerogenerador basado en el modelo creado por Hugh Piggott. Modelo económico, el cual puede ser realizado por cualquiera que tenga acceso a herramientas convencionales e idoneidad técnica, en la tabla N°10.4 se detallan sus especificaciones.

ESPECIFICACIONES	GENERADOR EOLICO
Potencia Nominal	700 W
Número de Hélices	3
Diámetro Hélice	1,25 metros
Material Hélice	Cedro
Voltaje de la Batería	12 V
Velocidad del viento nominal	10 m/s
Sistema de control	Timón Mecánico. Posicionamiento y Protección Automática
Alternador	2 Rotores de 12 imanes c/u y 1 estator de 9 bobinas
Peso	SIN ESPECIFICAR
Precio	\$15000 aprox..

Tabla 10.4. Especificaciones para el generador eólico de 700W nominales

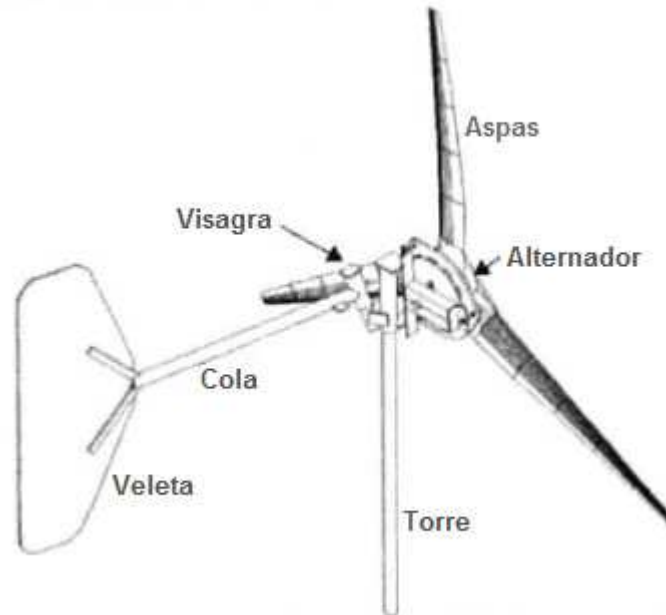
#### 10.6- Principio básico de funcionamiento de aerogenerador modelo Hugh Piggott

##### 10.6.1- Descripción mecánica.

Un aerogenerador está conformado por dos elementos principales: un rotor compuesto por un eje y la o las palas que es accionado por el viento, y un generador que se

mueve por arrastre del rotor. El generador suele ser un alternador conectado directamente al eje de rotación.

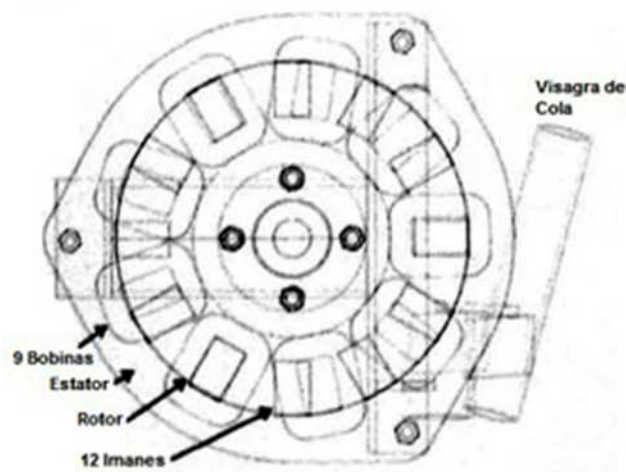
El aerogenerador se orienta hacia el viento mediante una veleta acoplada a la cola, de la misma forma que los molinos de viento rurales. Se soporta mediante una torre cuya altura depende de las condiciones del viento en el lugar de instalación, la figura N°10.1 nos presenta una estructura básica de un generador eólico de eje horizontal.



**Figura 10.1. Estructura básica de un Generador Eólico de eje horizontal.**

Cuando el viento incide de manera frontal a una velocidad mayor de 3 m/s, se produce el giro de sus aspas, cuya velocidad dependerá de la del viento. Mediante este movimiento rotacional, el alternador transforma la energía mecánica en energía eléctrica.

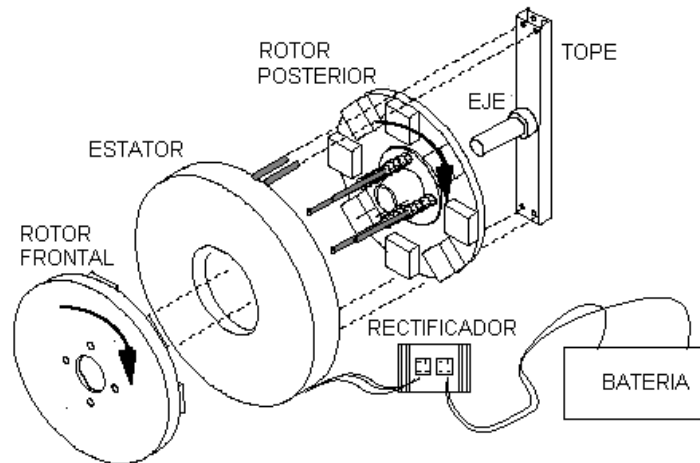
Como se ve en la figura N°10.2, el alternador está formado por un estator que contiene 9 bobinas y dos rotores conformados por 12 imanes de neodimio cada uno (24 en total). Este modelo de generador, produce una tensión de frecuencia y amplitud variable, por lo que no se puede utilizar directamente. La energía eléctrica producida es acondicionada mediante un sistema electrónico para luego ser almacenada en baterías.



**Figura 10.2. Vista frontal del conjunto Rotor, Estator y eje del Generador Eólico.**

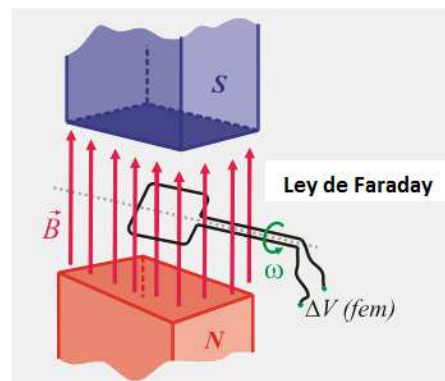
### 10.6.2- Generación de corriente eléctrica

Un generador eólico es un dispositivo que convierte energía eólica en energía eléctrica. Esta energía eléctrica es generada debido al movimiento que produce el rotor con respecto al estator. Tanto la disposición del rotor como la del estator es la que se puede observar en la figura 10.3. Vale destacar que en el aerogenerador de 700W realizado, se utilizó un rotor frontal y otro rotor posterior de 12 imanes cada uno, los cuales se colocan en el eje móvil que rotará en función al movimiento de las aspas al ser empujadas por la fuerza del viento



**Figura 10.3. Disposición del Rotor, Estator y eje del Generador Eólico, y conexasión con el Rectificador y luego con la Batería.**

Este movimiento hará girar a los imanes del rotor en torno a su eje y se producirá una variación del flujo magnético en las bobinas del estator. Según la ley de Faraday, al provocar un flujo magnético variable en el tiempo en una o más espiras se induce una diferencia de potencial a lo largo de la misma y en consecuencia una corriente cuyo valor depende de esa diferencia de potencial. Por lo tanto se induce una corriente que circulará en por las bobinas, por lo que entre los bornes aparece una diferencia de potencial  $\Delta V$  (fuerza electromotriz inducida), figura N°10.4. De esa manera se obtiene energía eléctrica a partir de la energía cinética del viento. Esta corriente generada por el campo magnético será corriente alterna, y es por ello que se deberá conectar las bobinas a un rectificador primero para luego cargar la o las baterías.



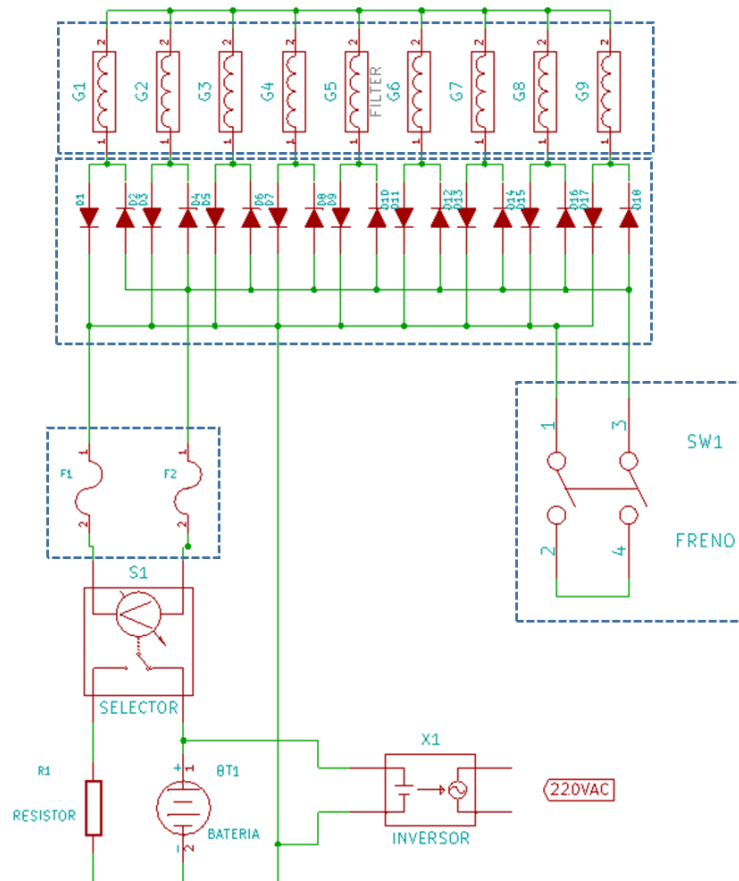
**Figura 10.4. Energía eléctrica generada a partir de la Ley de Faraday**

## 10.7- Desarrollo

### 10.7.1- Bloques y conexionado del sistema

#### 10.7.1.1- Conexionado general del sistema

La figura N°10.5 no presenta el conexionado del general aerogenerador.



**Figura 10.5. Diagrama del conexionado del aerogenerador.**

A continuación se detallan brevemente cada uno de los bloques involucrados.

#### 10.7.1.2- Rectificador

El bloque de rectificación es la parte del circuito electrónico encargada de convertir la señal eléctrica de corriente alterna a corriente continua. La señal de corriente alterna es la corriente procedente de las nueve bobinas en paralelo del estator con una tensión y una frecuencia variable en función de la velocidad de giro del rotor. Al tratarse de un rectificador no controlado compuesto por diodos, éstos conmutarán de manera natural forzados únicamente por la señal recibida del generador en cada uno de sus ánodos. La secuencia de conducción dependerá del diodo superior del circuito que tenga en su ánodo el mayor potencial positivo de todos y el diodo inferior que tenga en su cátodo el mayor potencial negativo, es decir, el de mayor y menor valor respectivamente ofrecida por el generador. En ese instante el resto de los diodos estarán polarizados en inversa con lo que no conducirán. Como en cada instante de

tiempo va a ser una fase diferente la que tenga mayor amplitud de señal positiva y negativa, en cada instante de tiempo va a conducir una pareja de diodos diferente.

#### **10.7.1.3- Fusibles**

Se conectan los puentes de diodos a las borneras y ahí el extremo (+) a un fusible de 63 Amper y de ahí a otras borneras que serán la alimentación de las baterías. Esto es para proteger al circuito de alimentación contra sobrecargas y cortocircuitos. Además entre las borneras de alimentación y el regulador se coloca otro fusible para proteger el regulador contra corrientes excesivas [112].

#### **10.7.1.4- Freno del molino**

Si bien el diseño planteado está orientado al tener un bajo mantenimiento, es recomendable frenar el aerogenerador para realizar una inspección general mecánica y eléctrica, una vez al año. Llegado el momento de esta situación, sería necesario una forma de detener el movimiento de las aspas, para evitar que se dañen. En este momento es donde encontramos la necesidad de tener un freno. El freno de nuestro generador eólico se encuentra compuesto solo por una llave térmica bifásica, conectada a cada a la línea de continua. Cuando decidimos frenar las aspas, accionaremos la llave, poniendo así en cortocircuito el generador. El principio de funcionamiento en el que se basa en la Ley de Faraday-Lenz [113]. La gran ventaja que este tipo de freno posee es que en el no intervienen fenómenos mecánicos, sino que todo se procede debido a campos magnéticos, evitando así los desgastes producidos por frenos mecánicos.

#### **10.7.1.5- Selector o Controlador de Carga y Resistor de Descarga**

Los controladores o reguladores de carga son equipos que controlan el voltaje y la corriente de un generador eólico, entregados al banco de baterías. Es un elemento indispensable en cualquier instalación que se precie, pues cuando la batería está totalmente cargada, el aporte de corriente solo puede acortar su vida. Si el generador entrega demasiada potencia el problema sería todavía peor al poder incluso estropearla. Por ello los reguladores conectan un elemento de carga (una resistencia convenientemente refrigerada) que absorbe toda la energía sobrante, si el generador entrega más de la necesaria o la batería estuviera totalmente cargada.

Los cables de conexión entre el aerogenerador y el regulador deben ser de la sección apropiada para no sufrir pérdidas, y será naturalmente de diámetro mayor a mayor potencia del generador, pero también deben ser más importantes cuanto más distancia exista entre el generador y el regulador, para evitar de esta manera pérdidas de rendimiento debidos a la resistencia eléctrica del cable (efecto Joule) [114].

#### **10.7.1.6- Baterías**

Las baterías almacenan la electricidad que genera el alternador del aerogenerador, para poderla suministrar posteriormente en cualquier momento. La capacidad de la batería se define como la cantidad de energía eléctrica que puede almacenar y se mide en Amperios-hora. Si por ejemplo una batería es de 100 Amperios-hora, esta podrá entregar una corriente eléctrica de 100 Amperios durante una hora, o de 50 Amperios durante 2 horas, o de 25 Amperios durante 4 horas, o de 5 Amperios durante 20 horas [115].

### 10.7.1.7- Inversor

El inversor es un dispositivo encargado de convertir la señal eléctrica de corriente continua proveniente de la batería a corriente alterna de 220V. Por no ser parte de los objetivos planteados inicialmente, no se abordara el tema.

### 10.7.1.8- Jabalina

Si bien no está representado en el diagrama general de conexiones, en cualquier ámbito que involucre temas relacionados con la electricidad y/o electrónica, siempre es necesario tener una buena conexión al punto de referencia desde el cual tomamos las diferencias de potenciales, principalmente por motivos de seguridad a los equipos que conforman el sistema, y a los usuarios y personas que estén en las cercanías de él.

### 10.7.2- Aspas, pivote y rotor

Para más información acerca del desarrollo y construcción de las aspas, del eje pivotante y del rotor (todos ellos partes mecánicas), por favor dirigirse al manual de construcción de aerogeneradores, brindado por la fundación "500RPM" [112].

La figura N°10.6 nos muestra una imagen de la construcción del rotor.



Figura 10.6. Construcción del rotor.

### 10.7.3- Estator

Este componente del sistema es el encargado de generar la energía eléctrica, a partir de los efectos magnéticos generados por los rotores que se encuentran próximos a él. Se encuentra compuesto básicamente por 9 bobinas dispuestas en forma circular, e interconectadas de una forma particular. Todo el conjunto de bobinas se rellena con una resina epoxi, con una forma generada a partir de un molde de madera, construido previamente. Los principales puntos a tener en cuenta en su realización son:

- Selección de las bobinas
- Construcción de las bobinas
- Conexión de las bobinas



- Protección de las bobinas
- Mediciones sobre las bobinas

### 10.7.3.1- Selección de las bobinas

El diámetro del conductor o los conductores utilizados para la construcción de las bobinas dependerá casi exclusivamente de la tensión de trabajo y del diámetro de las aspas de nuestro aerogenerador. El diámetro adecuado para cada una de las posibilidades se encuentra volcado en la tabla N° 10.5. Como podemos ver en ella, el diámetro del conductor correspondiente para un generador eólico con aspas de 2400 mm de diámetro y para una tensión de trabajo de 12V es de 1,7 mm, con un número de 73 vueltas por bobina.

### 10.7.3.2- Construcción de las bobinas

Para la realización de las 9 bobinas que componen a nuestro estator, se utilizó un bobinador construido especialmente para este proyecto, dado que estas debían tener formas y medidas específicas. Para más información acerca de esta herramienta [112]. Cada una de las bobinas tiene un espacio interno de forma rectangular, con las mismas medidas que los imanes de neodimio utilizados para la construcción de los rotores. Estas medidas son 25 mm de ancho y 45 mm de alto, con un espesor de ellas es de 11mm. La cantidad de vueltas y el grosor del alambre conductor se especifican en la tabla N°10.5.

Diámetro de la turbina [mm]		1200	1800	2400	3000	3600	4200
Peso del conductor [Kg]		1,5	2,6	3	5	5	7
12V	Diámetro [mm]	1,4	1,4	1,7	0,8	1,7	1,7
	Nº de vueltas	76	130	73	90	80	100
24V	Diámetro [mm]	1,06	1,5	2x1,5	2x1,6	2x1,8	2x1,8
	Nº de vueltas	140	110	45	55	37	45
48V	Diámetro [mm]	0,75	1,06	2,5-1,5	1,6	1,8	1,8
	Nº de vueltas	270	220	90	110	75	90

**Tabla 10.5. Diámetro de conductor, cantidad de vueltas y peso de las bobinas en función del diámetro de las aspas y de la tensión de trabajo**

### 10.7.3.3- Cálculo teórico de las bobinas

Los dos parámetros más importantes que se deben tener en cuenta al momento de diseñar y construir una bobina son la inductancia [116] que esta presenta, y su factor de mérito Q [117]. Como hemos visto en la tabla 10.5, el diámetro del conductor correspondiente para un generador eólico con aspas de 2400 mm de diámetro y para una tensión de trabajo de 12V es de 1,7 mm, con un número de 73 vueltas por bobina. Las dimensiones de las bobinas se eligieron de forma que el espacio interno de aire tenga un área mayor que el área de cada imán, para que así pudiera concatenar el

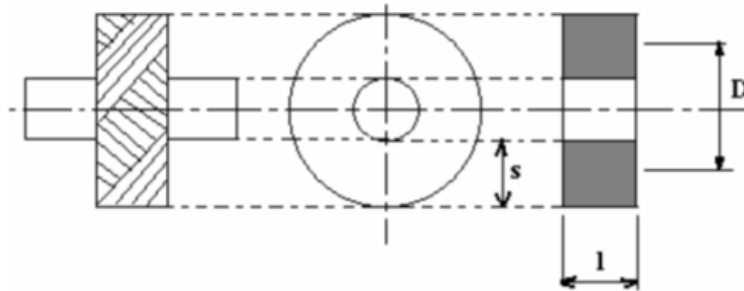
mayor flujo magnético posible. En base a esto se bobinó el inductor y se midieron las siguientes dimensiones:

- a = 25mm
- b = 45mm
- l = 12mm
- s = 15mm (aproximadamente)
- D = 57mm (aproximadamente)
- N = 73 vueltas

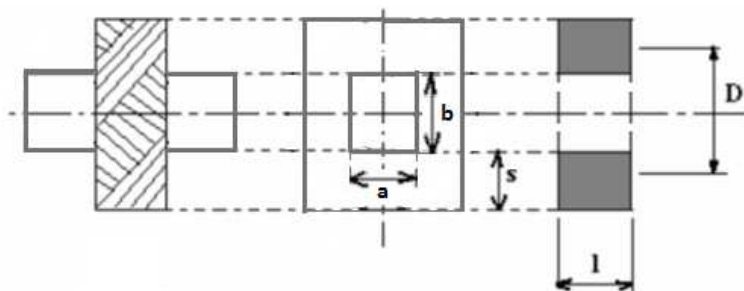
Dichas medidas pueden verse representadas en las figuras 10.7 y 10.8.

A partir de estos parámetros se puede obtener el valor teórico aproximado de la inductancia de cada bobina aplicando la fórmula de Wheeler para inductores multicapa sin núcleo provista por la cátedra de Tecnología Electrónica, cuya expresión N°10.1 es:

$$L [\mu H] = \frac{0,2 \cdot D^2 \cdot N^2}{7,6 \cdot D + 22,8 \cdot l + 25,4 \cdot s} \quad D, l \text{ y } s \text{ en [cm]} \quad (10.1)$$



**Figura 10.7. Dimensiones de Inductores Multicapa circulares.**



**Figura 10.8 - Dimensiones de Inductores Multicapa rectangulares.**

Para poder aplicar la fórmula de Inductores Multicapa se deberá hacer una equivalencia en las dimensiones de las bobinas. En tal caso haremos que la sección que ocupa la bobina rectangular sea el mismo al de una bobina circular.

Al proceder a realizar el cálculo se obtiene el siguiente resultado:

$$L = \frac{0,2 \cdot 5,7^2 \cdot 73^2}{7,6 \cdot 5,7 + 22,8 \cdot 1,2 + 25,4 \cdot 1,5} = 318,33 \mu H$$

Si bien este resultado se obtuvo para una bobina circular, se debe considerar que las dimensiones de una rectangular no presentarán grandes diferencias con la anterior, ya que la distancia entre el centro y la capa gruesa que contiene las espiras será similar en ambos casos porque se trata de la misma bobina que se la deforma levemente para adaptarla a la forma correspondiente para la medición.

Por otro lado se utilizó un software llamado MiscEI (Miscellaneous Electronic Calculations) [118], el cual permite calcular, entre muchas otras cosas, la inductancia de una bobina multicapa cuadrada. Los datos ingresados son:

- N= 73 espiras
- L= 12mm de longitud
- Lado (side) = 30mm de lado del cuadrado correspondiente al núcleo de aire. Las dimensiones en la bobina son a=25mm y b=45mm y para el cálculo se utilizó un valor intermedio de 30mm de lado.
- Diámetro del alambre = 1,7mm
- Espesor del bobinado = 15mm

El resultado que arrojó el software es una inductancia de 263uHy, figura N°10.9, lo cual es un valor distinto al calculado con la fórmula de Wheeler. Esto es lógico ya que este software contempla la forma cuadrada de la bobina, pero sin embargo presentará diferencias con el valor real ya que esta en realidad tiene forma rectangular. Aun así presenta una buena aproximación para estimar la inductancia.

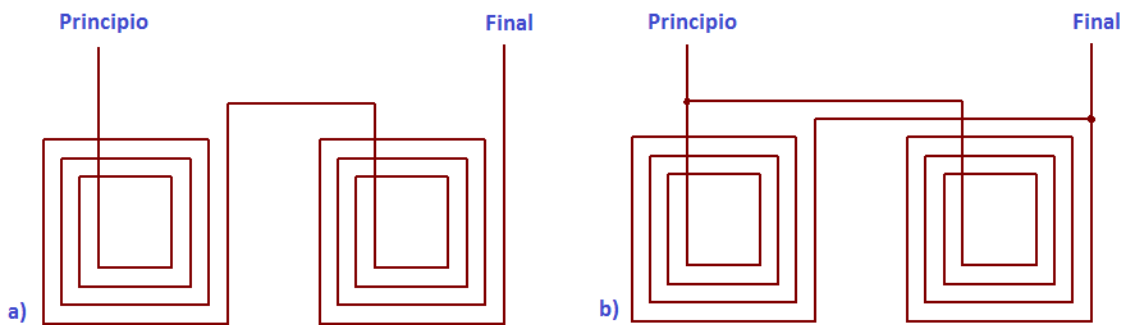
The screenshot shows the 'Miscellaneous electronic calculations' software window. The 'Air coil' section is active, and the 'Kind' is set to 'Multi layer square (Simple)'. The calculated inductance is 263 μH. The diagram shows a square coil with dimensions L (length) and Side (width), and a winding thickness. The input parameters are: Turns (N): 73, Length (L): 12m, Side: 30m, Wire diameter (mm): 1.7m, and Winding thickness: 15m. The output parameters are: Coil length: 12 mm, Avg. side: 45 mm (30 mm .. 60 mm), Winding thickness: 15 mm, Wire length: 15.3 m, Resistance: 110 kohm (copper wire at 20°C), and Estimated mass: 312 μg (of copper wire). A note states: 'An air coil does not always have a core filled with air, but must have a core filled with something non-magnetic (Plastic or wood is fine). All calculations assumes non-magnetic wires (copper is fine). Remember that all measurements (except wire diameter) are in meters! Use a m after the number to get mm, i.e. 0.01 and 10m are the same value. To enter values in inch use value\*inch, i.e. 3 inch is entered as 3 \*inch (For feet use ft)'.

Figura N° 10.9: Cálculo de la inductancia con el software “MiscEI”

### 10.7.3.3- Conexión de las bobinas

Al momento de realizar el conexionado de las bobinas, nos encontramos con distintos tipos de posibilidades, de las cuales solo una nos permitirá cumplir con las especificaciones que decidimos para nuestro generador eólico. Cada bobina produce un voltaje, pero el voltaje del estator depende de cómo se conecten dichas bobinas

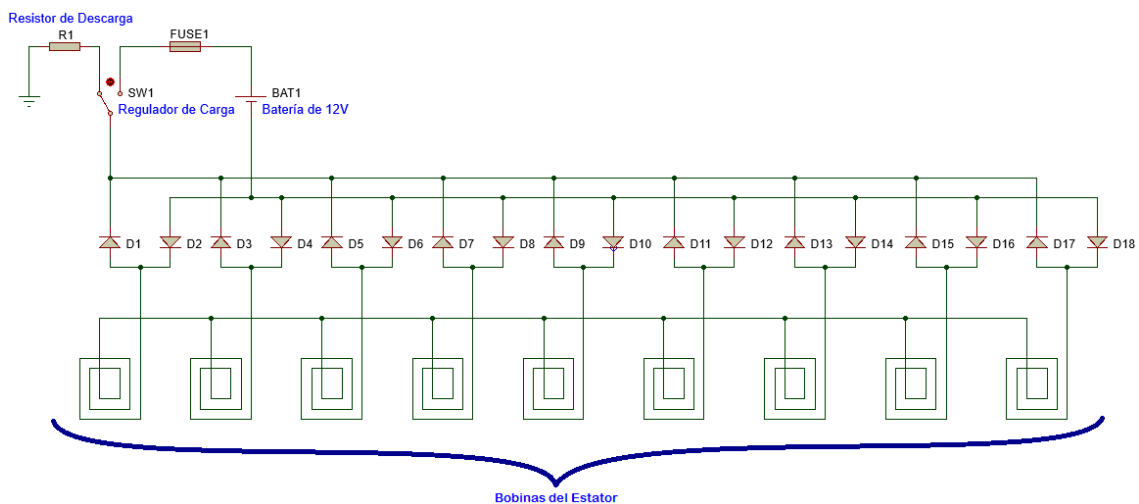
entre sí. El conexionado puede ser en Serie o en Paralelo como se muestra en la figura N°10.10:



**Figura 10.10. a) Conexionado de dos bobinas en Serie. b) Conexionado de dos bobinas en Paralelo.**

Como se puede observar, en la conexión en serie el final de una bobina se conecta con el principio de la siguiente. Por lo tanto los voltajes de las bobinas se suman para producir un voltaje doble entre el principio y el final del par. Por otro lado en una conexión en paralelo ambos principios de las bobinas se conectan entre sí al igual que los finales, y así se produce un mismo voltaje resultante en ambas bobinas. El principal objetivo de la conexión en paralelo es que en este caso las bobinas podrán llevar el doble de corriente. El inconveniente de realizar conexionados en Paralelo es el hecho de que el voltaje de cada bobina sea ligeramente diferente. Estas diferencias conducen a corrientes parásitas entre las bobinas que desperdician energía. En cambio las conexiones en Serie (o también llamadas “conexión en Estrella”) producen tres fases del mismo voltaje pero en tiempos diferentes. Su principal ventaja es que producen una salida más constante.

Debido a las ventajas y desventajas de los conexionados expuestas en el párrafo anterior, la mayoría de los estatores poseen un conexionado en Serie. Sin embargo los estatores de 12V conectados en serie conllevan un difícil manejo del hilo esmaltado para realizar las bobinas. Se puede observar en la tabla 10.5 que a medida que el voltaje generado es menor, se utiliza un diámetro de alambre de cobre mayor para el bobinado. Es por ello que resulta más eficiente realizar una conexión en Paralelo para los estatores de 12V. Además, para evitar las pérdidas por corrientes parásitas se utilizarán dos diodos por cada bobina, figura N° 10.11, es decir que el rectificador se encontrará ubicado en la turbina.

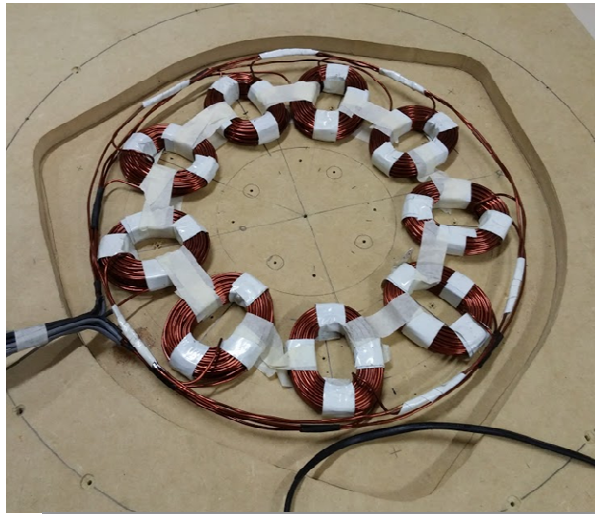


**Figura N°10.11. Esquema del conexionado en Paralelo para las bobinas del estator.**

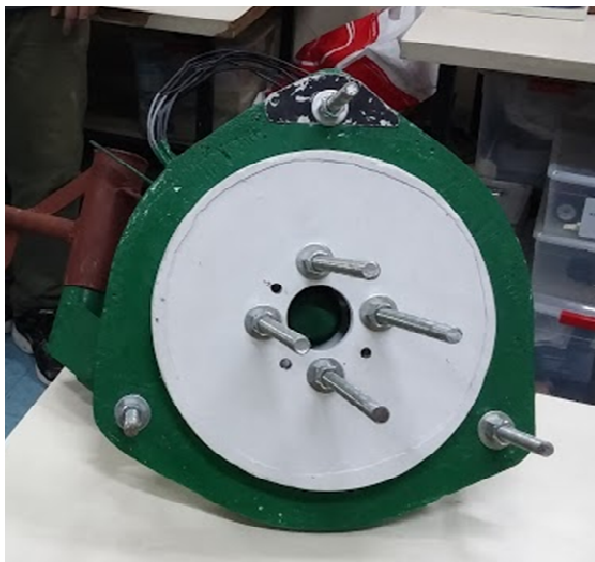
#### 10.7.3.4- Protección de las bobinas

El armado de las bobinas y su correspondiente conexionado no tendrían sentido si no pudiésemos generar con ellas una estructura sólida, que las proteja y las sostenga firmemente, para poder acomodarlas en medio de los dos rotores, y así generar la energía eléctrica deseada. Por este motivo es que utilizamos la misma resina que empleamos en los rotores para proteger los imanes. Gracias a la ayuda de un molde fabricado en fibro fácil, las bobinas fueron colocadas en su posición, y todo el conjunto fue rellenado con ella [112].

La figura 10.12 y 10.13 nos muestra la construcción del estator y el sistema estator-rotor armado.



**Figura 10.12. Construcción del estator.**



**Figura 10.13. Sistema Rotor- Estator.**

### 10.7.3.5- Mediciones sobre las bobinas

En la tabla N°10.6 se muestran las mediciones realizadas del Q y L de una bobina. Se utilizó para ello 2 instrumentos: el Qúmetro HP 4285A y el LCR Meter ELC-130.

Frecuencia [kHz]	Factor de Mérito Q	Inductancia [uHy]
530	35,06	179,683
540	35,29	179,979
550	35,47	180,255
600	36,45	181,754
650	37,51	183,136
700	38,01	184,627
750	38,76	186,270
800	39,15	188,102
850	39,73	190,127
900	39,81	192,418
950	39,73	194,973
1000	40,16	197,678
1030	40,18	199,424
1050	40,19	200,751
1070	40,17	202,055
1080	40,2	202,595
1100	40,16	204,114
1200	39,74	211,979
1300	38,58	221,194
1400	38,03	232,077
1500	36,82	245,741
1600	35,31	262,497
1700	33,62	282,523
1800	31,81	307,423
1900	29,74	342,761
1920	29,31	351,548

**Tabla N°10.6. Valores medidos en una de las bobinas mediante el Qúmetro “Hp Modelo 4285A”**

Las figuras N°10.14 y 10.15 muestra los graficos obtenidos de las mediciones realizadas.

Finalmente utilizamos el instrumento LCR Meter Digital Modelo ELC-130, el cual permite la medición de la inductancia a una frecuencia fija de 1kHz y obtuvimos un  $L=150\mu\text{Hy}$ .

Si bien cálculo teórico no se corresponde con los valores medidos se debe considerar que la fórmula de Wheeler es aproximada y si bien contempla la existencia de varias capas de arrollado, en la práctica el bobinado no es ideal porque puede presentar disparidades, como por ejemplo espiras cruzadas y capas que se entrecruzan entre sí.

Aún con estas diferencias se puede contemplar como conclusión de la medición, que la inductancia de las bobinas se encontrará en el orden de los 100uHy.

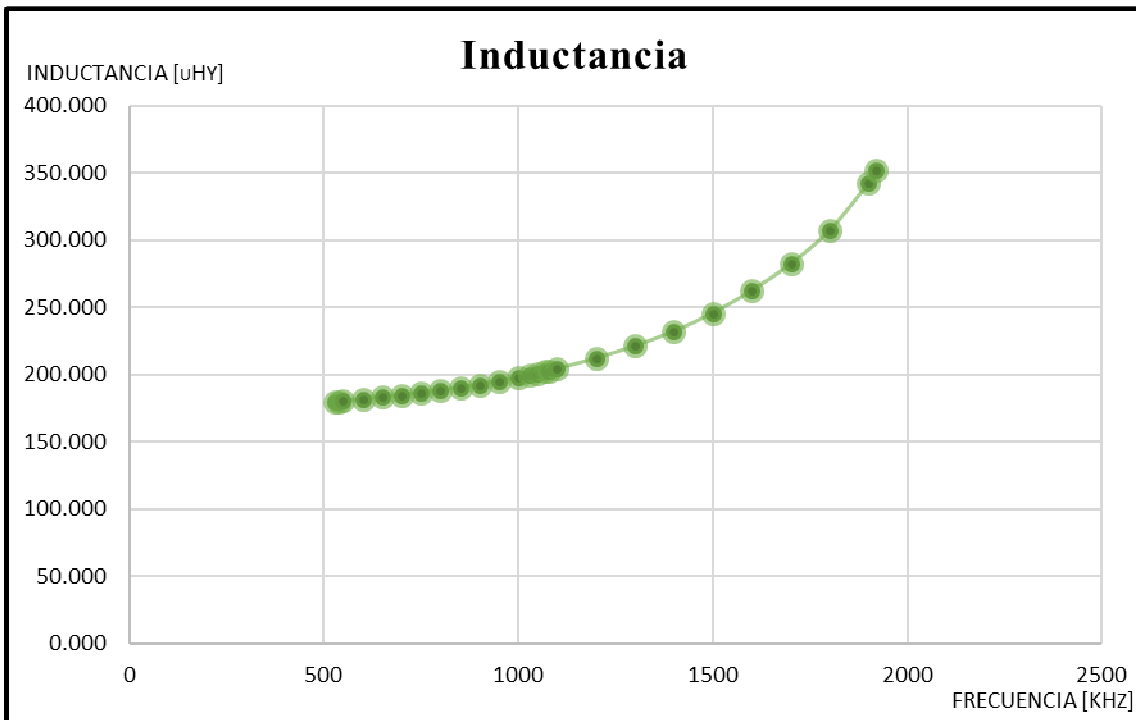


Figura 10.14. Representación de la Inductancia en función de la Frecuencia según las mediciones realizadas con el Qúmetro “Hp Modelo 4285A”

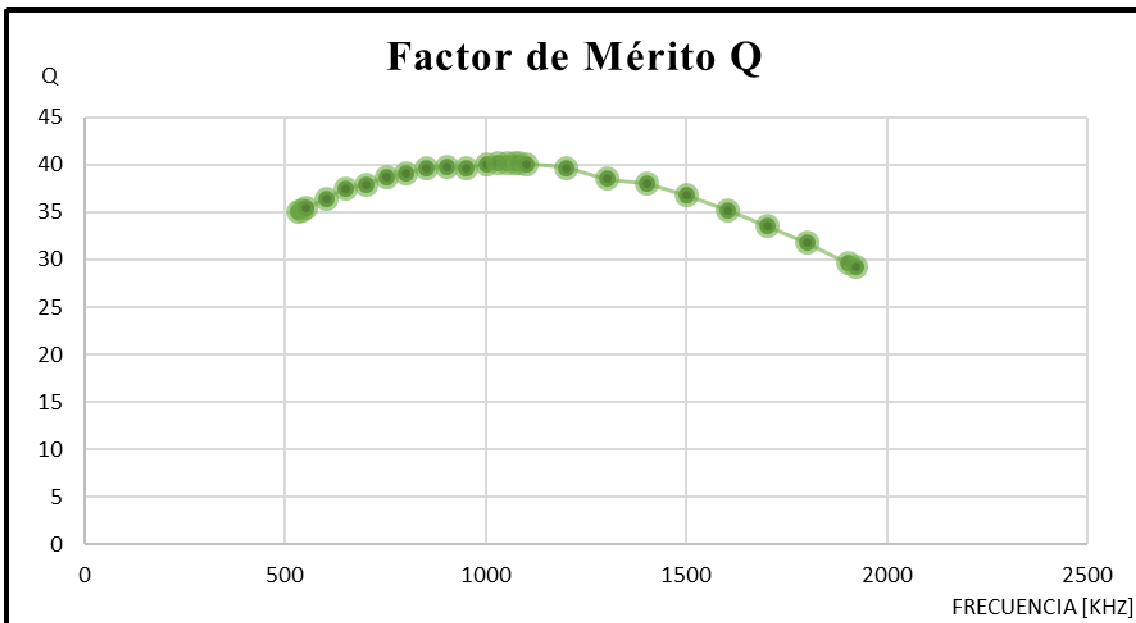


Figura 10.15. Representación del Factor de Mérito en función de la Frecuencia según las mediciones realizadas con el Qúmetro “Hp Modelo 4285A”

#### 10.7.4- Controlador de Carga

Como mencionamos anteriormente en el informe, la tarea principal del controlador de carga es la de proteger el banco de baterías, permitiendo o no el paso de la energía

eléctrica generada por el aerogenerador hacia ellas, dependiendo del estado de carga en el que se encuentran.

### 10.7.4.1- Diseño esquemático

La figura 10.16, nos presenta el diagrama esquemático de nuestro regulador de carga.

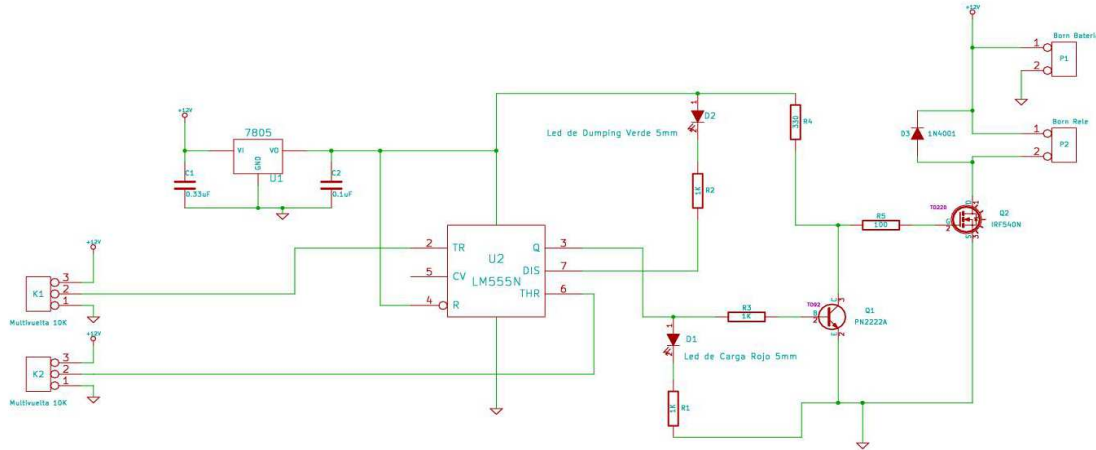


Figura 10.16. Circuito del Regulador de Carga.

### 10.7.4.2- Principio de funcionamiento

El componente principal de nuestro regulador se encuentra conformado por un circuito integrado NE555 [119]. Su diagrama en bloques se puede ver en la figura N°10.14.

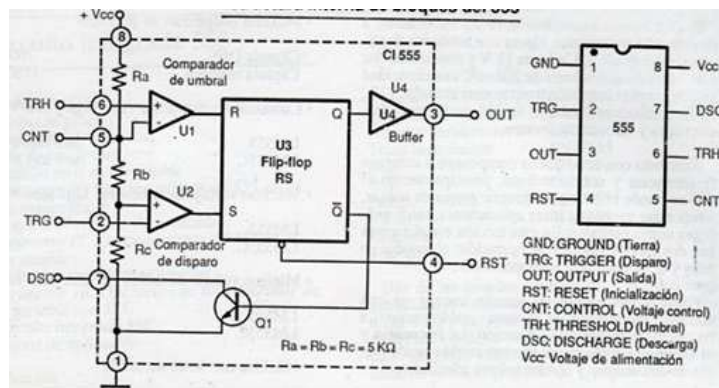


Figura 10.14. Diagrama en bloques del IC NE555

A través de dos potenciómetros multivuelta se establecen las tensiones en los pines TRH (THR en circuito) y TRG (TR en circuito), en función de las cuales el NE555 generará la conmutación entre el circuito de carga y la carga ficticia.

## 10.8- Ensayos y experimentos

Existe un gran número de pruebas y ensayos que se deberían hacer para comprobar fehacientemente el cumplimiento de todas las especificaciones con las que fue diseñado. El abanico es amplio, y cubre áreas que van desde el funcionamiento mecánico de la cabeza, el rendimiento aerodinámico de las aspas, hasta la eficiencia eléctrica de todo el circuito.



Esto incluye pruebas destructivas para estimar las partes más frágiles o sujetas a desgaste mecánico, la instalación en un lugar permanente para conocer la evolución del rendimiento a lo largo de un año, y el armado de dispositivos auxiliares para realizar estas pruebas. Por motivos pura y exclusivamente económicos, no es imposible afrontar el costo de estas pruebas.

De todas formas, es adecuado mencionar y desarrollar los principales ensayos que deberían ser realizados, que nos permitan verificar aproximadamente las especificaciones.

#### **10.8.1- Ensayo de rendimiento**

Este es el principal ensayo que debería ser realizado, dado que nos permitiría observar al generador eólico en condiciones extremas, teniendo un completo control sobre las variables que afectan su trabajo. Para realizar este tipo de pruebas es necesario contar con un túnel de viento, con posibilidades de controlar la velocidad de viento, pudiendo así especificar cuándo comienza a generar energía, así como la velocidad a la cual alcanza la potencia máxima para la cual fue diseñado (potencia nominal). Además, nos permite trazar una gráfica de la energía generada versus la velocidad del viento. A esto se le conoce como prueba de rendimiento de conversión de energía cinética en energía eléctrica.

#### **10.8.2- Ensayo de plegado**

En el mismo túnel de viento en el cual se realizaría la prueba de rendimiento, se busca activar las protecciones del prototipo, elevando la velocidad del viento por sobre la cual se alcanza la potencia nominal, en este instante es donde se puede observar en funcionamiento el sistema de plegado mecánico, diseñado para proteger el circuito eléctrico. Este ensayo es simplemente nombrado a modo informativo ya que nuestro aerogenerador no permite el plegado.

#### **10.8.3- Ensayo de posicionamiento**

Para la realización de este ensayo deberemos encontrar una forma práctica, mediante la cual direccionar el viento que incide sobre el aerogenerador, para corroborar el correcto funcionamiento del timón, encargado de llevar a las aspas a una posición perpendicular al viento.

#### **10.8.4- Pruebas realizadas**

La prueba consistió en acoplar mediante una planchuela solidaria a un eje concéntrico, el centro de rotación del generador a un taladro de alto torque. En este caso, el taladro utilizado es un Dowen Pagio modelo 9993868, de alto torque y uso profesional.

El taladro cuenta con un limitador de velocidad que permite variarla entre 0 y 550 RPM, con lo que nos aseguramos no superar el máximo permitido. Las vibraciones producidas por las desviaciones de los centros, son absorbidas por el operario del taladro, evitando así el esfuerzo mecánico del prototipo y la máquina. Este taladro cuenta además con doble empuñadura, una tipo pistola, y otra posterior tipo "D", lo que ayuda a sujetarlo mejor.

La planchuela cuenta con un sistema de correderas que permite ajustar la posición relativa de ambos centros, lo cual también ayuda a disminuir las vibraciones, la figura 10.17 muestra en acople implementado.



**Figura 10.17 - Acople de taladro y generador**

Una vez ajustado todo, se procede a hacer girar el prototipo, poniendo en funcionamiento el taladro, al principio en la velocidad más baja posible, y a medida que la inercia se valla rompiendo, se aumenta la velocidad gradualmente hasta alcanzar la máxima permitida. De esta forma, logramos obtener una curva de tensión generada, versus velocidad de rotación.

La medición se realizó haciendo uso de un osciloscopio OWON SDS5032E, del cual se utilizó un solo canal, conectado al par de conductores de salida del generador. La escala vertical se ajustó en 2v/div y la escala temporal en 1s/div. Al variar la velocidad de rotación a medida que transcurre el tiempo, se puede suponer que la escala horizontal tiene un equivalente en unidades de RPM.

En la figura N° 10.18 se puede observar como aumenta la tensión entre extremos del generador, sin carga conectada, a medida que se incrementa la velocidad de rotación. La velocidad máxima a la que se lo hizo rotar fue de entre 230 y 270 RPM, y se obtuvo una tensión que varía entre 16v y 18v.

Al no poder medir de forma directa las revoluciones a las que giraba el taladro, se procedió a realizar las mismas de forma indirecta, a partir de un video capturado durante la realización del ensayo. Para ello se marcó un extremo de la planchuela y se procedió a verificar el tiempo que tardó en realizar un giro completo. La planchuela tarda un total de aproximadamente 28 cuadros de video en realizar un giro completo, lo que con una velocidad de captura de 120FPS da un equivalente a 0.23 segundos por vuelta, lo que resulta por ende en una velocidad de 257RPM.

Luego se procedió a conectarle una carga, y ver como esto afecta el rendimiento del mismo. La carga conectada fue una resistencia calefactora de un calo-ventor, la cual tiene una resistencia en frio de aproximadamente 35Ω. Se puede ver en la figura N°10.19 que la máxima tensión en DC alcanzada es de alrededor de 16v, y el resultado es muy similar al de la medición en vacío, con lo cual la carga no produce una variación importante en la tensión en DC a la salida del rectificador.

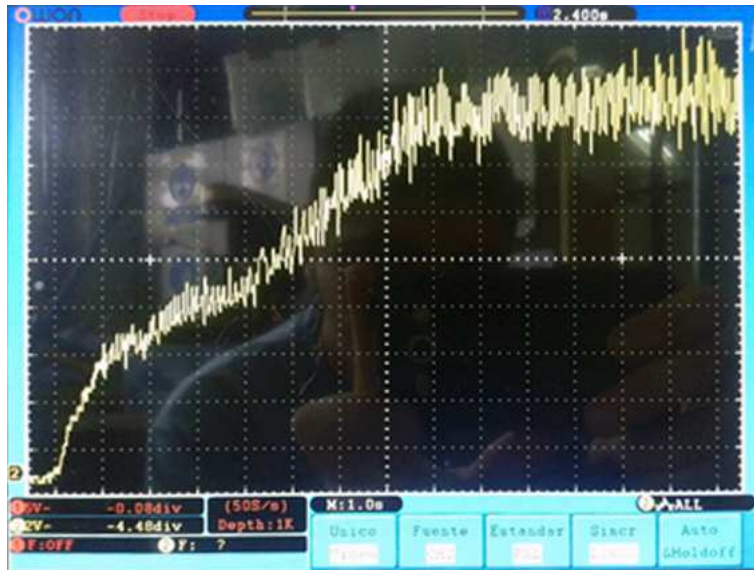


Figura 10.18 - Tensión versus RPM en vacío

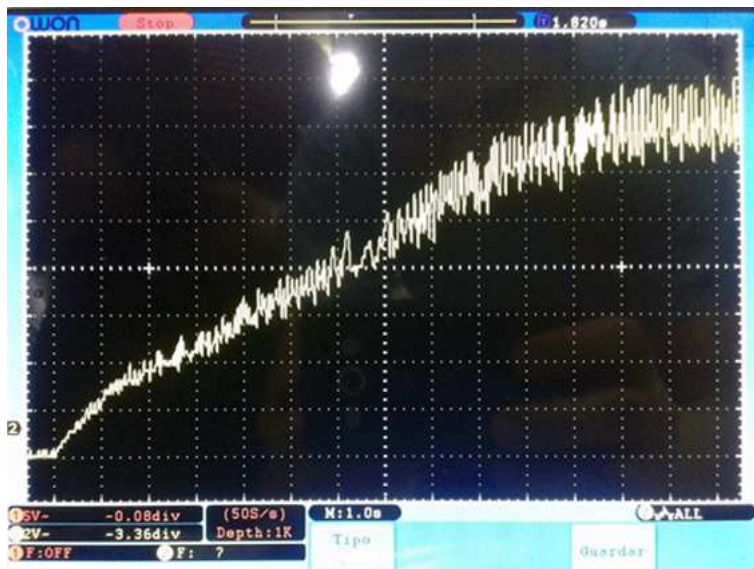


Figura 10.19 - Tensión versus RMP cargado

### 10.8.5- Otras posibles pruebas

Otra posibilidad consiste en acoplar mecánicamente un motor al eje del generador, no siendo necesaria la presencia de las aspas. De esta manera, controlando las revoluciones del motor, podemos realizar una curva que caracterice la potencia generada acorde a la velocidad de rotación del mismo.

Las dificultades que afronta este tipo de prueba consiste en que se debe lograr un acoplamiento perfecto entre ambos sistemas, un balance minucioso del generador, y un control estricto sobre la velocidad del motor, ya que cualquier deriva de los parámetros estándares, podrían ocasionar la ruptura del prototipo.

## 10.9- Conclusiones

Analizando las mediciones realizadas, podemos concluir que el generador tiene un rendimiento acorde a lo estipulado. De esta forma, nos permitimos proyectar nuevos horizontes en el desarrollo de este tipo de dispositivos. Las figura 10.20 nos muestra el prototipo finalizado.

Los procesos de fabricación pueden ser mejorados, abaratando costos y tiempos de fabricación, y a su vez perfeccionar las técnicas de modelado en resina, el diseño de las aspas, el material de las mismas, etc.

Además consideramos, se puede mejorar la electrónica asociada. Por ejemplo, incursionar en el desarrollo de inversores para elevar la tensión a 220VCA, profundizar en el estudio de las baterías de descarga profunda, y cambiar la utilización de un controlador de carga por un regulador de carga para mejorar el proceso de carga de la batería o banco de baterías.



**Figura 10.20. Prototipo finalizado.**

## **11 - Sistema de transferencia automática para grupos electrógenos**

### **11.1 - Introducción**

La electrificación rural, muchas veces no rentable para las empresas de distribución de energía eléctrica, posee una gran importancia ya que permite la integración de los sectores rurales al desarrollo económico nacional, frena la migración rural-urbana y mejora el nivel socio cultural de los habitantes entre otras. Es por estos motivos que a nivel país existen políticas de electrificación rural en distintos países de Latinoamérica [85][86]. En particular en Argentina podemos nombrar el proyecto de electrificación rural para el desarrollo pecuario del departamento de 25 de mayo, en la provincia de San Juan [87], el cual está bajo el Programa de Servicios Agrícolas Provinciales (PROSAP), dependiente del ministerio de Agricultura, Ganadería y Pesca.

Dentro de estos programas está contemplado el uso de energías renovables, eólica, fotovoltaica, biomasa, etc., ejemplos del uso de energía eólica se pueden ver en el proceso de electrificación del espacio rural del Partido de Tandil, proceso llevado adelante por la Cooperativa Rural Eléctrica de Tandil-Azul Limitada (CRETAL) [88], así como ejemplos de energía fotovoltaica para la agricultura y el desarrollo sostenible se pueden ver en [89].

Toda fuente de energía renovable debe contar con un sistema de almacenamiento de energía, el cual le permita almacenar la energía generada y disponer de ella cuando sea necesaria. En función del dimensionamiento de dicho sistema se podrá contar, por un determinado tiempo, con energía aun cuando las condiciones ambientales no sean las favorables para la generación. Pero como las condiciones ambientales están fuera de nuestro alcance es importante contar con un sistema de respaldo, en este caso un grupo electrógeno.

En muchas aplicaciones productivas, donde el personal no tiene acceso permanente a las instalaciones, y se necesita un suministro de energía constante, es necesario contar con un sistema de transferencia automática. El mismo, permite el constante monitoreo de los parámetros de la fuente de suministro, y se encarga de hacer de forma automática la transferencia de energía a la carga, ya sea de la línea al grupo electrógeno, en caso de corte de luz o disminución de la energía almacenada, o del generador a la línea, en caso contrario.

### **11.2 - Estudio de Mercado**

Luego de realizada una búsqueda de este tipo de sistemas en el mercado, nos encontramos con productos de empresas multinacionales como el Sistema de Control Modular EPIC de Caterpillar [97], los automatismos BA y UA de Schneider Electric [98] y el sistema SENTRON ATC5300 de Siemens [99]. Empresas más pequeñas ofrecen una gran variedad de implementaciones como el sistema AUT-MP12 de la empresa Electra Molins [100] que se basa en un módulo programable con tres microprocesadores, el sistema de transferencia automática con contactores de la empresa Velázquez Ingenieros Asociados LTDA basado en el sistema de transferencia automática ITACMC [101] y el sistema de la empresa Control Para la Industria basado en PLC Omron de la familia CJ1[102]. También se han encontrado gran variedad de trabajos realizados en esta temática [103][104][105], en cuyos casos el sistema implementado se basa en un controlador de lógica programable (PLC).

Los sistemas presentados pueden ser separados en dos grandes grupos: los sistemas dedicados y los sistemas basados en PLC. Los primeros suelen estar pensados para cubrir las demandas estándar de la aplicación, siendo poco flexibles a adaptarse a

casos particulares del usuario. Los sistemas basados en PLC, si bien son más flexibles, esta flexibilidad suele estar asociada a la incorporación de módulos extra, lo que genera un aumento de los costos finales del sistema.

Debido a estos motivos, se planteó el desarrollo de un sistema de transferencia automática para grupos electrógenos de bajo costo y flexible.

### 11.3 - Descripción general

El trabajo que se presenta es parte de un sistema más complejo que involucra la medición de parámetros de consumo eléctrico, de generación de energías renovables y climáticos, su transmisión a un nodo central, su concentración y posterior transmisión a un servidor para su análisis.

El sistema toma muestras de cada una de las fases (R S y T) tanto de línea como del grupo electrógeno para controlar los valores de tensión y frecuencia con la finalidad de mantener los valores dentro de los parámetros deseados. Si alguno de los valores de la fuente de energía aplicada a la carga no está dentro los parámetros preestablecidos conmuta automáticamente a la otra fuente. Los valores muestreados son a la vez transmitidos a una estación colectora [90] para su retransmisión y análisis. La figura 11.1 muestra el diagrama en bloques del sistema, mientras que la figura 11.2 muestra el diagrama de conexión del sistema a la línea eléctrica.

El sistema de transferencia automática cuenta con los siguientes módulos:

- Medidor de parámetros de suministro de línea: con el cual se monitorea la tensión de línea y las corrientes de la carga.
- Medidor de parámetros de suministro de grupo: con el cual se monitorea la tensión de grupo y las corrientes de la carga.
- LCD: es el módulo donde se visualizan los datos.
- Módulo Serie/USB: es el encargado de la comunicación Serie/USB.
- Módulo GSM/GPRS: es el encargado de hacer la comunicación GSM y GPRS con el usuario.
- Relé contactores: es el módulo que hace la habilitación de cada contactor.
- Relé solenoide, arranque, alarma: es el que habilita el paso de combustible, hace el arranque y hace sonar una alarma externa en caso de haber algún error.

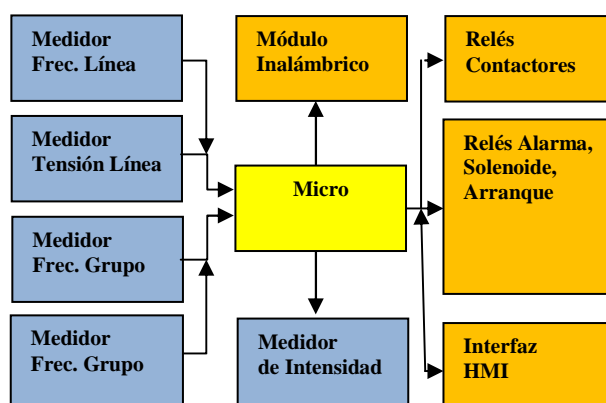
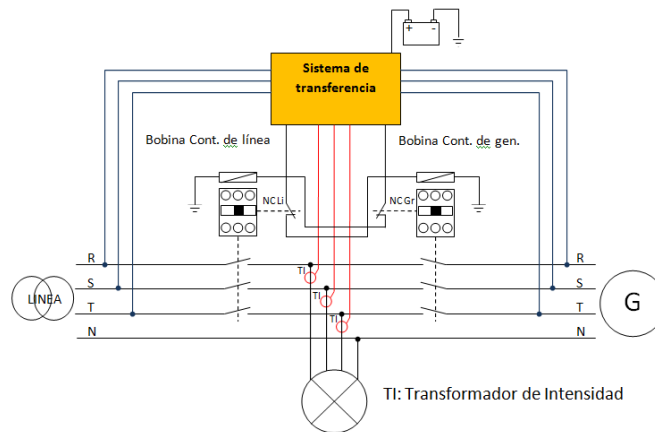


Figura N° 11.1 - Diagrama en bloques del sistema



**Figura N°11.2 - Diagrama de conexión del sistema a la línea eléctrica.**

#### 11. 4 - Hardware

Como base de nuestro desarrollo se tomó un microcontrolador basado en el núcleo Cortex-M4 de ARM. El producto seleccionado fue el STM32F407ZET6 de ST [91]. La elección se fundamentó en:

- La disponibilidad de conversores analógicos digitales para la medición de parámetros.
- Gran número de interfaces seriales (I2C, UART, SPI), que posibilita la comunicación con sensores y distintos módulos OEM.
- Interfaz Ethernet
- Capacidad de procesamiento adecuada para trabajar con una pila TCP/IP.

Otro factor que influyó en la selección de este microcontrolador, es la posibilidad de que el sistema pueda actuar como estación colectora y si bien el desarrollo actual de la estación colectora [90] se basó en el ATSAM3X y se había planteado una evolución natural al ATSAM4E, dicha opción se descartó por la disponibilidad y costo de los microcontroladores y los kits de evaluación en el mercado local. Siendo el STM32F4xx y el ATSAM4E de similar desempeño se consideró adecuado el cambio.

Los bloques medidores de parámetros de suministro de línea y grupo, encargados de monitorizar las tensiones y corrientes de los mismos, están formado por el integrado ATM90E36A [92] de la firma ATMEL. Dichos dispositivos se encuentran en la parte de alta tensión del sistema. Para aislar los mismos del microcontrolador se utilizó los integrados de aislación digital de Silabs, Si8641 [93], que poseen aislación de 5kV y una velocidad de transferencia máxima de 150Mbps. La medición de corriente se realiza a través de un transformador de corrientes, AC1020 de la firma Talema [94].

En la figura 11.3 se puede ver una parte del circuito impreso del sistema. En el mismo se pueden apreciar tres bloques, dos de ellos correspondientes a la parte de alta tensión, en los cuales se encuentran los ATM90E36A y el tercero al de baja tensión, en el cual se encuentra el microcontrolador. Entre ambos bloques de pueden apreciar la existencia de circuitos integrados, los cuales son los aisladores Si8641.

#### 11.5 - Entorno de desarrollo integrado(IDE)

Para la realización del proyecto se utilizó el IDE de IAR Workbench ARM versión 7.3.

## 11.6 - Firmware

El firmware del microcontrolador del sistema se encuentra estructurado de la siguiente forma: inicialización de periféricos, configuración de los dispositivos de medición (ATM90E36A) y un ciclo infinito en el cual se verifica los valores eléctricos y actúa en función de ellos. La acción a realizar depende del estado del grupo electrógeno (Detenido, Arranque, Marcha), esta se representa en el diagrama de flujo en el bloque “Alarmas”, figura 11.4. A continuación se presentan los valores medidos en el display de la interfaz hombre máquina (HMI), se verifica la recepción de algún comando de configuración y se envían los datos por el puerto serie para ser presentados en el software de PC. En la figura N° 11.2 se ve el diagrama de conexión del sistema a la línea eléctrica.

La principal virtud del firmware implementado radica en la rápida y transparente configuración de los parámetros del dispositivo de medición. El ATM90E36A cuenta con más de 100 registros para la lectura de los valores y la configuración de todas las funcionalidades del mismo [92].

La figura 11.5 presenta una pantalla del software desarrollado, a través de la cual se puede configurar la ganancia y el offset para cada una de las fases y el neutro conectados al ATM90E36A, en este caso particular los asociados a la medición del grupo electrógeno. Una vez configurados los valores y presionado “enviar”, los mismos son almacenados en los registros correspondientes del dispositivo de medición.

A continuación se detalla el código de inicialización de los dispositivos de medición:

### **// Reset chip Generator /Line**

```
Atm90e36a_SetDataG(0x0000,0x789A);  
Atm90e36a_SetDataL(0x0000,0x789A);  
Delay_us(120000);
```

### **// Lee los parametros de los chips medidores**

```
Atm90e36a_Read_ParamG();  
Atm90e36a_Read_ParamL();  
// Configura los chips medidores
```

```
Atm90e36a_Conf_updateG();  
Atm90e36a_Conf_updateL();
```

### **//Setea los parametros de los chips medidores**

#### **// Calc Checksum 2**

```
Atm90e36a_SetDataG(0x0050,0x5678);  
ATM90E36A_GEN.HarmStart=Atm90e36a_GetDataG(0x0050);  
ATM90E36A_GEN.CS2=atm90e36a_checksum(&ATM90E36A_GEN.POffsetAF,6);  
Atm90e36a_SetDataG(0x0057,ATM90E36A_GEN.CS2);  
ATM90E36A_GEN.CS2=Atm90e36a_GetDataG(0x0057);  
ATM90E36A_GEN.SysStatus0=Atm90e36a_GetDataG(0x0001);  
Atm90e36a_SetDataG(0x0050,0x8765);
```

#### **// Calc Checksum 3**

```
Atm90e36a_SetDataG(0x0060,0x5678);
```

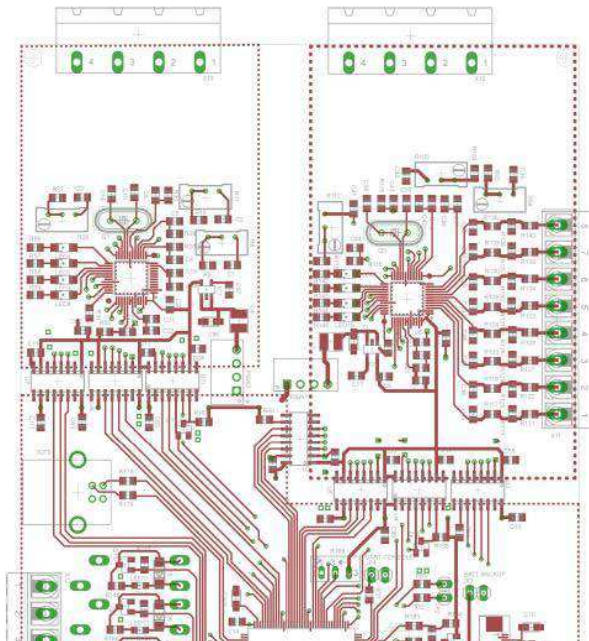


```

ATM90E36A_GEN.AdjStart=Atm90e36a_GetDataG(0x0060);
ATM90E36A_GEN.UgainA=0x8B2F;
Atm90e36a_SetDataG(0x0061,ATM90E36A_GEN.UgainA);
ATM90E36A_GEN.UgainB=0x8B2F;
Atm90e36a_SetDataG(0x0065,ATM90E36A_GEN.UgainB);
ATM90E36A_GEN.UgainC=0x8B2F;
Atm90e36a_SetDataG(0x0069,ATM90E36A_GEN.UgainC);
ATM90E36A_GEN.IgainA=0x1AF5;
Atm90e36a_SetDataG(0x0062,ATM90E36A_GEN.IgainA);
ATM90E36A_GEN.IgainB=0x1AF5;
Atm90e36a_SetDataG(0x0066,ATM90E36A_GEN.IgainB);
ATM90E36A_GEN.IgainC=0x1AF5;
Atm90e36a_SetDataG(0x006A,ATM90E36A_GEN.IgainC);
ATM90E36A_GEN.IgainN=0x1AF5;
Atm90e36a_SetDataG(0x006D,ATM90E36A_GEN.IgainN);
ATM90E36A_GEN.CS3=atm90e36a_checksum(&ATM90E36A_GEN.UgainA,14);
Atm90e36a_SetDataG(0x006F,ATM90E36A_GEN.CS3);
ATM90E36A_GEN.CS3=Atm90e36a_GetDataG(0x006F);
ATM90E36A_GEN.SysStatus0=Atm90e36a_GetDataG(0x0001);
Atm90e36a_SetDataG(0x0060,0x8765);

```

En la figura N° 11.3 se presenta la imagen parcial del circuito impreso.



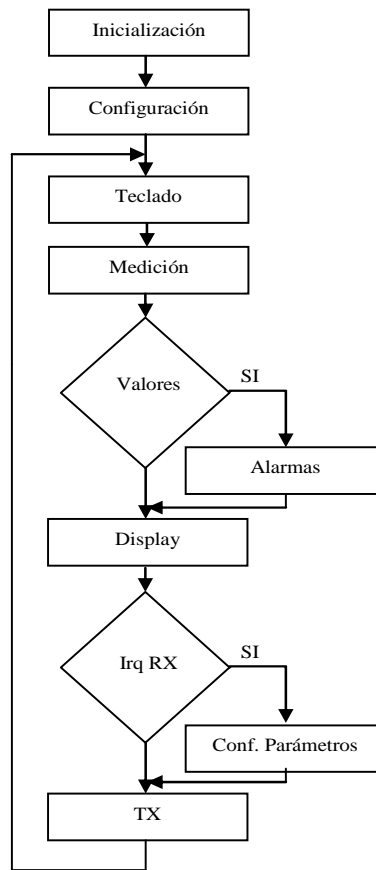
**Figura N°11.3 - Imagen parcial del circuito impreso.**

### 11.7 – Ensayos

Finalizados los chequeos de hardware, firmware y software el sistema debe ser calibrado para la correcta medición de tensiones y corrientes. Esto se puede realizar por medio del software de PC, previamente nombrado, y/o de unos potenciómetros en la placa.

La calibración, tanto para la corriente como para la tensión, se debe realizar de a una línea por vez, es decir que si se quiere calibrar la medición de tensión de la fase A en el grupo se debe cortocircuitar la fase B y C. La misma se debe llevar a cabo inyectando una tensión conocida y comparándola con la medición que entrega el

equipo en la pantalla de la PC. Desde la pantalla se puede modificar tanto la ganancia como el offset para lograr que ambas mediciones coincidan en su valor. Si la calibración no se lograra mediante el software de PC, el cual modifica los parámetros de los chips medidores, se puede ajustar la ganancia vía los potenciómetros ubicados en la placa para tal fin.



**Figura N° 11.4 - Diagrama de flujo del firmware.**

Se realizaron mediciones de tensión y corriente luego de la calibración de los dispositivos, los valores obtenidos se pueden ver en la tabla N°1.

Para la calibración y las posteriores mediciones se utilizaron la pinza UT208 y el multímetro digital UT71A ambos de la firma UNI-T [95][96].

Se observó que los dispositivos son capaces de medir con muy buena exactitud los valores de entrada y que sobrepasaron las expectativas planteadas en el comienzo del proyecto, ya que no es necesaria tanta exactitud en la aplicación planteada. La tabla N°11.1 refleja los valores de tensión entregada y tensión medida y la figura N° 11.6 muestra una imagen del sistema.

### 11.8 - Conclusiones

Se ha logrado el desarrollo de un sistema capaz de medir con exactitud los valores eléctricos de la línea y los generados por un grupo electrógeno, y de realizar la conmutación entre ambos sistemas a fin de mantener el suministro de energía sobre la carga.

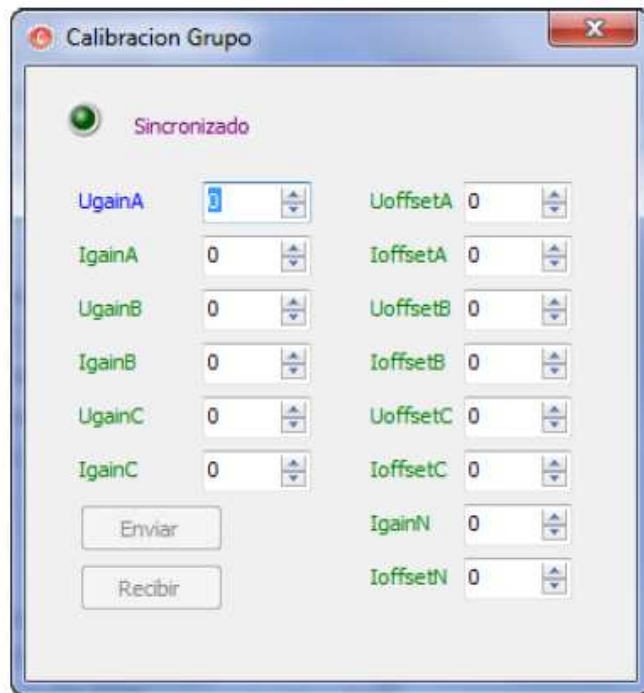


Figura N°11.5 - Pantalla de calibración.

Tabla N°1. Comparación de valores reales y medios.

<b>Tensión Entregada [V]</b>	<b>Tensión Medida [V]</b>
223,67	223,45
219,49	219,32
113,82	113,67
110,49	110,26
<b>Corriente Consumida[A]</b>	<b>Corriente Medida[A]</b>
3,678	3,669
7,902	7,89
10,615	10,601

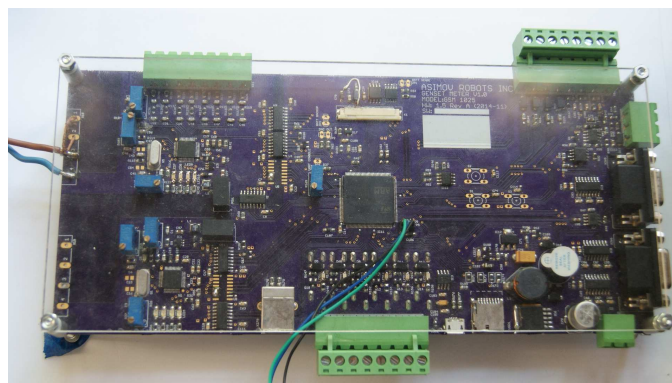


Figura N°11.6 - Imagen del sistema.

Para los prototipos desarrollados (dos) se realizó un gasto de aproximadamente U\$S 200 por unidad, costo el cual se estima puede llegar a menos de la mitad en producción. Siendo dicho costo un valor competitivo.

La exactitud de valores medidos se ha logrado gracias a la amigable interfaz de PC, que permite fácilmente configurar los parámetros del dispositivo de medición. La eficiencia de esta conmutación, es decir la estabilidad de la energía entregada a la carga, el tiempo de respuesta, la suavidad de la transición, no se han llegado a probar, quedando dichos ensayos como uno de los próximos pasos a realizar.

El sistema desarrollado posee un gran número de prestaciones que no se encuentran implementadas, como ser la comunicación Ethernet, la medición de parámetros asociados a la calidad de la señal eléctrica, el uso de un display inteligente como interfaz local. Se tiene como objetivo la pronta implementación de dichas características, lo que permitirá al sistema poder trabajar en redes eléctricas inteligentes.

## **12 - Evaluación del equipo y presupuesto.**

### **12.1- Periodo 2014.**

En líneas generales el grupo de trabajo se desempeño bien, pero se observó que los tiempos estipulados para las tareas asignadas han resultado no ser suficientes, en cierta medida esto se debe a que el grupo de trabajo se encuentra conformado por alumnos y docentes cuya dedicación está distribuida en varias actividades académicas. Es decir el grupo no cuenta con personal dedicado exclusivamente a la investigación, por lo tanto los avances son dependientes de las otras cargas académicas.

En lo referente al presupuesto, un parcial del mismo fue asignado a fines de octubre, por lo que al momento del presente informe no se ha podido hacer un uso del mismo. Se ha evaluado la participación en Iberchip 2015 (<http://www.ieee-lascas.org/lascas2015/iberchip/information.html>), el cual se realizara en la ciudad de Montevideo – Uruguay, lamentablemente se debió descartar la participación en el mismo debido a que los costos de traslado, hospedaje e inscripción resultaron muy elevados para el presupuesto asignado.

### **12.2- Periodo 2015.**

El problema de dedicación de los miembros del grupo se mantuvo. El caso puntual de los alumnos, involucrados inicialmente, uno de ellos tuvo que dejar el grupo y otro pudo participar muy poco. Con el fin de cumplir con los objetivos plantados se incorporo otro alumno al grupo, así como también se trabajó con varios grupos de alumnos de la cátedras de proyecto final y Tecnología electrónica orientando sus proyectos de cátedra a los objetivos de la investigación. Esto último repercutió positivamente en los objetivos del grupo. Dichos grupos de alumnos contaron con la asistencia de los miembros del grupo.

En lo referente al presupuesto, al no ser posible cumplir con la participación en el congreso internacional previsto (Iberchip 2015) los fondos se destinaron a otros dos de los rubros, insumos y equipamiento. Esto permitió la construcción del aerogenerador presentado y la compra de un PC de escritorio para el grupo de trabajo, lo cual redundó en una mejor organización del trabajo. Es importante destacar que los fondos solicitados inicialmente para el rubro viáticos, estaban adecuados pero al efectuarse el pago de los fondos destinados al proyecto en dos cuotas no fue posible afrontar los gastos. De esta experiencia surge que si se desea encarar un proyecto que involucre aspectos mecánicos y a la vez tener fondos disponibles para viáticos, es muy difícil de lograr con los fondos asignados.

## **13 - Transferencias**

### **13.1- Periodo 2014.**

“Plataforma de conexión de Redes Eléctricas Inteligentes a Internet de las Cosas” Canziani, Mónica; Gómez, Rodrigo; Lupi, Daniel; Nassipián, Verónica; Slawiski Javier; Turconi, Diego; Zaradnik, Ignacio, en Congreso Argentino de Sistemas Embebidos, CASE2014, Buenos Aires 13- 14 Agosto 2014 FIUBA. . ISBN 978-987-45523-2-7.

Conferencia “From Nanotech to IoT for Smart Factory” Daniel Lupi, Delegado por Argentina; en “The XX World MicroMachine Summit” Berlin, Germany; 10 – 13 May 2014.

### **13.2- Periodo 2015.**

Conferencia “Internet of Things as a Latin America Big Market” Daniel Lupi; in *SEMI South America Semiconductor Strategy Summit*; Rio de Janeiro, Brasil, 10-12 November 2015

“Sistema de transferencia automática para grupos electrógenos” Turconi, Diego; Agüero, Jorge; Campero, Ignacio; Farina, Juan; Lupi, Daniel; Zaradnik, Ignacio. Buenos Aires Congreso Argentino de Sistemas Embebidos (CASE 2015)-. ISBN 978-987-4552-3-4

“Desarrollo de un sistema fotovoltaico con seguimiento solar de dos ejes” Agüero, Jorge; Alfarano, Maximiliano; Cipollone, Federico; Gimenez, Ezequiel; Greto, Nicolás; Franco, Nicolás; Tantignone, Hugo; Lupi, Daniel; Zaradnik, Ignacio. Buenos Aires, Congreso Microelectrónica Aplicada 2015 (uEA 2015)-. ISBN 978-987-3806-24-7

“Diseño interfaz inalámbrica para microsistemas” Canziani, Mónica; Castellini Ariel; Lupi, Daniel; Oppedisano, Ruben; Slawiski, Javier; Turconi, Diego; Zaradnik, Ignacio. Buenos Aires, Congreso Microelectrónica Aplicada 2015 (uEA 2015)-. ISBN 978-987-3806-24-7

Conferencia sobre “Internet de las Cosas (IoT) y sus Aplicaciones en el Ámbito Agropecuario”; Zaradnik, Ignacio; en el Congreso Microelectrónica Aplicada 2015 (uEA 2015).

### **14- Agradecimientos**

Se agradece a: Agüero, Jorge; Campero, Ignacio; Fariña, Juan; Alfarano, Maximiliano; Cipollone, Federico; Giménez, Ezequiel; Greto, Nicolás; Franco, Nicolás; Ayllón, Rodrigo; Cervera, Julio; Muranot543, Emmanuel; Rios, Brian; Valles, Gabriela; Avido, Matías; Gagliardi, Leandro; Quintana, Ricardo; Rios, Sergio; Villar, Juan; Tantignone Hugo; Mariño, Nestor, alumnos y docentes que participaron en los distintos desarrollos realizados.

### **15 – Conclusiones**

El principal objetivo, lograr un sistema inteligente para la administración de la energía en establecimientos agrícolas a través del concepto de “Internet de las Cosas”, no llegó a implementarse completamente. Entre las causas de esto se pueden nombrar: que los integrantes tuvieron que armar dispositivos complejos y que sus dedicaciones están distribuidas en varias actividades académicas; este año se produjeron en alguno de ellos situaciones personales no esperadas y por último la discontinuidad de productos.

Sin embargo, los temas investigados y desarrollados para tal fin han logrado generar una base sólida de conocimiento en la temática, llegándose a profundizar en algunos de ellos más de lo esperado. Con lo cual, el objetivo de alcanzar experiencias suficientes para recomendar acciones y estrategias que apoyen el desarrollo y aplicación del concepto de “Internet de las Cosas” para la optimización y utilización de

los recursos renovables, y el objetivo de consolidar el conocimiento y fortalecer los recursos humanos en lo que respecta al manejo y aplicación de estas tecnologías informáticas, han sido logrados.

Ejemplo de esto último es la implementación práctica de varios prototipos de trabajo de sistemas de energía renovable, los que fueron ya descriptos previamente. Esto permitió conocer en detalle los costos de fabricación e implementación de dichos sistemas, sus elementos y la factibilidad de implementación en las distintas áreas de nuestro país.

Como trabajo futuro queda el perfeccionamiento de los sistemas de energía renovable implementados, desarrollo de una nueva microestación con nueva tecnología, el agregado de la medición de energía eléctrica y el análisis de estos datos por el software.

## 16 - Bibliografía

- [1] Tiempo de Juan, "San Juan, capital de la energía solar", <http://www.tiempodesanjuan.com/sanjuan/2012/1/2/juan-capital-energia-solar-4238.html>, última visita 12/12/2015.
- [2] INTI, "Nueva planta solar fotovoltaica en San Juan", <http://www.inti.gob.ar/renova/erSO/er25.php>, última visita 12/12/2015.
- [3] Revista Megavatios 395, "Otorgan crédito para parque fotovoltaico Cañada Honda III", [http://www.edigarnet.com/e\\_revistas/webmagazine/index.php?pub=megavatios&num=395](http://www.edigarnet.com/e_revistas/webmagazine/index.php?pub=megavatios&num=395), última visita 12/12/2015.
- [4] Télam, "Otorgan crédito de 11 millones de dólares para la construcción en San Juan del mayor parque solar fotovoltaico de la región", <http://www.telam.com.ar/notas/201402/52040-otorgan-credito-de-11-millones-de-dolares-para-la-construccion-en-san-juan-del-mayor-parque-solar-fotovoltaico-de-la-region.html>, última visita 12/12/2015.
- [5] Jose Luis Cavassasi, "¿Que es una bateria de ciclo profundo?", <http://www.cavadevices.com/archivos/FOLLETOS/BATERIAS%20CICLO%20PROFUNDO.pdf>, última visita 12/12/2015.
- [7] CNEA - Comité Nacional de Energía Atómica, "Síntesis del Mercado Eléctrico Mayorista", <http://www.cnea.gov.ar/Sintesis-mercado-electrico-mayorista>, última visita 12/12/2015.
- [8] Revista Ingeniería Eléctrica N°283. Diciembre 2013, "Octubre demando mas de 10.000 GWh", [http://www.editores-srl.com.ar/revistas/ie/283/%252Fsites/default/files/ie283\\_fundelec\\_octubre\\_demando\\_mas\\_de\\_10000\\_gwh](http://www.editores-srl.com.ar/revistas/ie/283/%252Fsites/default/files/ie283_fundelec_octubre_demando_mas_de_10000_gwh), última visita 12/12/2015.
- [9] Oscar Medina, "Smart Grids – The development of Smart grid pilot projects in Argentina", <http://ccap.org/assets/Oscar-Medina-Argentina-Smart-Grid-Pilots.pdf>, última visita 12/12/2015.
- [10] Oscar Medina- Secretaria de energía – República Argentina , "REDES INTELIGENTES ¿REALIDAD, UTOPIA O FUTURO?", [https://www.energia.gov.ar/contenidos/archivos/Reorganizacion/contenidos\\_didacticos/publicaciones/redes\\_inteligentes.pdf](https://www.energia.gov.ar/contenidos/archivos/Reorganizacion/contenidos_didacticos/publicaciones/redes_inteligentes.pdf), última visita 12/12/2015.
- [11] Manuel Rodriguez, "El futuro de la red eléctrica es Smart Grid", <http://revistadigital.inesem.es/gestion-integrada/futuro-red-electrica-smart-grid/>, última visita 12/12/2015.
- [12] Observatorio Industrial del Sector de la Electrónica, Tecnologías de la Información y Telecomunicaciones, "El sector de electrónica, tecnologías de la información y comunicaciones en españa",

- [http://www.minetur.gob.es/industria/observatorios/SectorElectronica/Actividades/2010/introducci%C3%B3n\\_ETIC.pdf](http://www.minetur.gob.es/industria/observatorios/SectorElectronica/Actividades/2010/introducci%C3%B3n_ETIC.pdf), última visita 12/12/2015.
- [13] ETP Smart Grids “National and regional Smartgrids initiatives in Europe”, <http://www.smartgrids.eu/Overview%20of%20National%20and%20Regional%20Technology%20Platforms%20in%20Europe.pdf>, última visita 12/12/2015.
- [14] Un balance de los consumos eléctricos de 2012. Revista Ingeniería Eléctrica N°274, pag. 90-93. Marzo 2013. [http://www.editores-srl.com.ar/revistas/ie/274/noticias\\_balance\\_consumos\\_electricos\\_2012](http://www.editores-srl.com.ar/revistas/ie/274/noticias_balance_consumos_electricos_2012), última visita 12/12/2015.
- [15] Un balance de los consumos eléctricos de 2013. Revista Ingeniería Eléctrica N°288, pag. 92-97. Julio 2014. [http://www.editores-srl.com.ar/revistas/ie/288/un\\_balance\\_de\\_los\\_consumos\\_electricos\\_de\\_2013](http://www.editores-srl.com.ar/revistas/ie/288/un_balance_de_los_consumos_electricos_de_2013), última visita 12/12/2015.
- [16] Guillermina P. Jacinto, María Luciana Nogar, “Electrificación rural, desarrollo territorial y pequeñas localidades. El caso de Tandil (provincia de Buenos Aires, Argentina)”, [dialnet.unirioja.es/descarga/articulo/3947070.pdf](http://dialnet.unirioja.es/descarga/articulo/3947070.pdf), última visita 12/12/2015.
- [17] Secretaria de energía – República Argentina, “Energías renovables”, [http://www.energia.gov.ar/contenidos/archivos/Reorganizacion/contenidos\\_didacticos/publicaciones/EnergiasRenovables.pdf](http://www.energia.gov.ar/contenidos/archivos/Reorganizacion/contenidos_didacticos/publicaciones/EnergiasRenovables.pdf), última visita 12/12/2015.
- [18] IDAE Instituto para la Diversificación y ahorro de Energía, “Ahorro y Eficiencia Energética en Instalaciones Ganaderas”, [http://www.idae.es/uploads/documentos/documentos\\_10330\\_Instalaciones\\_ganaderas\\_05\\_8ad73059.pdf](http://www.idae.es/uploads/documentos/documentos_10330_Instalaciones_ganaderas_05_8ad73059.pdf), última visita 12/12/2015.
- [19] IDAE Instituto para la Diversificación y ahorro de Energía, “Ahorro, Eficiencia Energética y Sistemas de Laboreo Agrícola”, [http://www.idae.es/uploads/documentos/documentos\\_10330\\_sistemas\\_laboreo\\_agricola\\_a\\_06\\_ebaa2e1f.pdf](http://www.idae.es/uploads/documentos/documentos_10330_sistemas_laboreo_agricola_a_06_ebaa2e1f.pdf), última visita 12/12/2015.
- [20] IDAE Instituto para la Diversificación y ahorro de Energía, “Ahorro y Eficiencia Energética en Invernaderos”, [http://www.idae.es/uploads/documentos/documentos\\_10995\\_Agr07\\_AyEE\\_en\\_invernaderos\\_A2008\\_9e4c63f5.pdf](http://www.idae.es/uploads/documentos/documentos_10995_Agr07_AyEE_en_invernaderos_A2008_9e4c63f5.pdf), última visita 12/12/2015.
- [22] Héctor Fernando Mattio, Fernando Tilca “Recomendaciones para mediciones de velocidad y dirección de viento con fines de generación eléctrica, y mediciones de potencia eléctrica generada por aerogeneradores”, [http://www.inti.gob.ar/cirsoc/pdf/accion\\_viento/Recomendaciones\\_Mediciones\\_de\\_Viento.pdf](http://www.inti.gob.ar/cirsoc/pdf/accion_viento/Recomendaciones_Mediciones_de_Viento.pdf), última visita 12/12/2015.
- [23] Raúl Righini, Hugo Grossi Gallegos, “Mapa de Energía Solar colectada anualmente por un plano inclinado. Un ángulo óptimo en la república Argentina”, [http://www.cab.cnea.gov.ar/ieds/images/2011/hyfuse\\_n\\_2011/trabajos/11-161.pdf](http://www.cab.cnea.gov.ar/ieds/images/2011/hyfuse_n_2011/trabajos/11-161.pdf), última visita 12/12/2015.
- [24] BARGACH, M.N., TADILI, R., DAHMAN, A.S. y BOUKALLOUCH, M. “Comparison of the performance of two solar heating systems used to improve the microclimate of agricultural greenhouses in Morocco”. Renewable Energy, 29: 1073–1083. 2004.
- [26] B. van Campen, D. Guidi y G. Best, “Energía solar fotovoltaica para la agricultura y desarrollo rural sostenibles”, Documento de Trabajo sobre Medio Ambiente y Recursos Naturales, No.3 FAO, Roma, 2000, <http://www.fao.org/uploads/media/Solar%20photovoltaic%20for%20SARD%20ES.pdf>, última visita 12/12/2015.
- [27] Ambito Financiero, “Se desplomó 92% superávit comercial (sigue alta importación de combustibles)”, <http://www.ambito.com/noticia.asp?id=733439>, Marzo 2014, última visita 12/12/2015.
- [28] La Nación, “Una cuenta que crece: el millonario negocio de la importación de energía”, <http://www.lanacion.com.ar/m1/1570155-una-cuenta-que-crece-el-millonario-negocio-de-la-importacion-de-energia>, Abril 2013, última visita 12/12/2015.



- [29] La Razón, “En 2013, importación record de energía”, [http://www.larazon.com.ar/actualidad/importacion-record-energia\\_0\\_532800018.html](http://www.larazon.com.ar/actualidad/importacion-record-energia_0_532800018.html), Enero 2014, última visita 12/12/2015.
- [30] Indec, “Intercambio comercial Argentino”, [http://www.indec.gov.ar/uploads/informesdeprensa/ica\\_08\\_14.pdf](http://www.indec.gov.ar/uploads/informesdeprensa/ica_08_14.pdf), última visita 12/12/2015.  
[http://www.indec.gov.ar/informesdeprensa.asp?id\\_tema\\_1=3&id\\_tema\\_2=2&id\\_tema\\_3=40](http://www.indec.gov.ar/informesdeprensa.asp?id_tema_1=3&id_tema_2=2&id_tema_3=40)
- [31] Edenor, “Equipos de medición para tarifa 1”, [http://www.edenor.com.ar/cms/SP/CLI/COM/INF\\_COM\\_tarifa1.html](http://www.edenor.com.ar/cms/SP/CLI/COM/INF_COM_tarifa1.html), última visita 12/12/2015.
- [32] Elster, “Argentina mira hacia las redes inteligentes”, [http://www.elster.com.ar/es/noticias\\_y\\_eventos\\_1158.html](http://www.elster.com.ar/es/noticias_y_eventos_1158.html), Enero del 2013, última visita 12/12/2015.
- [33] ECYT-AR, “Energías renovables en Argentina”, [http://cyt-ar.com.ar/cyt-ar/index.php/Energ%C3%ADas\\_renovables\\_en\\_Argentina](http://cyt-ar.com.ar/cyt-ar/index.php/Energ%C3%ADas_renovables_en_Argentina), última visita 12/12/2015.
- [34] ATMEL, “AVR2130: Lightweight Mesh Developer Guide”, [http://www.atmel.com/images/atmel-42028-lightweight-mesh-developer-guide\\_application-note\\_avr2130.pdf](http://www.atmel.com/images/atmel-42028-lightweight-mesh-developer-guide_application-note_avr2130.pdf), última visita 12/12/2015.
- [35] <http://www.zigbee.org/Standards/Downloads.aspx>
- [36] ATMEL, “AVR2050: BitCloud Developer Guide”, [http://www.atmel.com/Images/Atmel-8199-BitCloud-Developer-Guide\\_User-Guide\\_AVR2050.pdf](http://www.atmel.com/Images/Atmel-8199-BitCloud-Developer-Guide_User-Guide_AVR2050.pdf), última visita 12/12/2015.
- [37] <http://www.contiki-os.org/index.html>, última visita 12/12/2015.
- [38] ATMEL, “Low Power 700/800/900 Mhz Transceiver for IEEE802.15.4-2006, IEEE802.15.4c-2009, Zigbee, 6LoWPAN, and ISM Applications”, <http://www.atmel.com/images/doc8168.pdf>, última visita 12/12/2015.
- [39] Pai H. Chou and Sehwan Kim, “Techniques for Maximizing Efficiency of Solar Energy Harvesting Systems”, <http://www.ece.uci.edu/~chou/pdf/chou-icmu10.pdf>, última visita 12/12/2015.
- [40] Hector Fernando Mattio, Fernando Tilca, “recomendaciones para mediciones de velocidad y dirección del viento con fines e generación eléctrica y medición de potencia eléctrica generada por aerogeneradores”, Octubre 2009 [http://www.inti.gob.ar/cirsoc/pdf/accion\\_viento/Recomendaciones\\_Mediciones\\_de\\_Viento.pdf](http://www.inti.gob.ar/cirsoc/pdf/accion_viento/Recomendaciones_Mediciones_de_Viento.pdf), última visita 12/12/2015.
- [41] EISON OLAYA ARBOLEDA, LUIS OCTAVIO GONZÁLEZ SALCEDO, “FUNDAMENTOS PARA EL DISEÑO DE BIODIGESTORES”, <http://www.bdigital.unal.edu.co/7967/4/luisoctaviogonzalezsalcedo.20121.pdf>, última visita 12/12/2015.
- [42] <https://www.metzoo.com>, última visita 12/12/2015.
- [43] <http://www.tyniot.com/metzoo-la-solucion-argentina-habilitadora-de-iot/>, última visita 12/12/2015.
- [44] Sebastian Davidovsky, “Dos desarrollos argentinos que aprovechan la tecnología para ayudar a los agricultores”, Diario la Nación, Mayo 2105. <http://www.lanacion.com.ar/1790816-dos-desarrollos-argentinos-aprovechan-la-tecnologia-para-ayudar-a-los-agricultores>, última visita 12/12/2015.
- [45] <http://boosteragtech.com/products.html>, última visita 12/12/2015.
- [46] Honeywell, “ViewPoint 2.3 Online Help”, Julio 2012.
- [47] <http://ubidots.com/about-ubidots.html>, última visita 12/12/2015.
- [48] <http://www.digi.com/cloud/digi-device-cloud>, última visita 12/12/2015.
- [49] <http://www.telit.com/products-and-services/iot-platforms/iot-portal/>, última visita 12/12/2015.
- [50] Energizar, “Seguidor Solar” [http://www.energizar.org.ar/energizar\\_desarrollo\\_tecnologico\\_seguidor\\_solar\\_que\\_es.html](http://www.energizar.org.ar/energizar_desarrollo_tecnologico_seguidor_solar_que_es.html), última visita 12/12/2015.

- [51] <http://www.weather-guide.com/ciudad-clima-es/Buenos-Aires-tiempo-clima.html>, última visita 12/12/2015.
- [52] Aurelio, Morales, Tantignone, Zaradnik, “Módulo de LEDs para iluminación pública”, II Congreso de microelectrónica aplicada, Septiembre 2011.
- [53] José Manuel Palomar-Carnicero, Pedro Casanova-Peláez, Fco. Alberto Díaz-Garrido, Fernando Cruz-Peragón, Rafael López-García, “NUEVO SISTEMA DE SEGUIMIENTO SOLAR DE DOS EJES BASADO EN EL MECANISMO DE BIELAMANIVELA” <http://www.revistadyna.com/Articulos/Ficha.aspx?idMenu=26e5f45a-7fdd-456b-8088-fad6a8390778&Cod=4344>, última visita 12/12/2015.
- [54] Asociación Argentina de Luminotecnia “Iluminación – Luz, Visión, Comunicación, Tomo 1 y 2”
- [55] Energía ERCAM, “Seguimiento solar”, [www.ercam.es/index.php/es/seguimiento-solar](http://www.ercam.es/index.php/es/seguimiento-solar), última visita 12/12/2015.
- [56] <http://ingemecanica.com/ingenieria/proyectos/proyecto106.pdf>
- [57] Javier Ródenas, “Ventajas de un seguidor solar” <http://www.terra.org/categorias/articulos/ventajas-de-un-seguidor-solar>, última visita 12/12/2015.
- [58] H. Magaña Castañeda, J. Ramírez, A. Romero Arguelles, “Diseño de un seguidor solar para eficientar la captación de la energía solar en los paneles fotovoltaicos”, Instituto tecnológico superior de Arandas, México, Octubre 2010.
- [59] Alberto Escudero Pascual, “Sistemas de energía solar fotovoltaica para equipos de telecomunicaciones. Introducción”, [http://www.it46.se/courses/solar/materials/es/IT46\\_es\\_energia\\_solar\\_introduccion.pdf](http://www.it46.se/courses/solar/materials/es/IT46_es_energia_solar_introduccion.pdf), última visita 12/12/2015.
- [60] Vitron energy, “Baterías de Gel y AGM”, <http://www.victronenergy.com.es/upload/documents/Datasheet%20-%20GEL%20and%20AGM%20Batteries%20-%20rev%2007%20-%20ES.pdf>, última visita 12/12/2015.
- [61] eNGineering, “Seguimiento del punto de máxima potencia y su funcionamiento”, <http://renewableengineering.blogspot.com.ar/2010/04/seguimiento-del-punto-de-maxima.html>, última visita 12/12/2015.
- [62] Electroimpulso Ingeniería, “Regulador para paneles Solares 12V-10Amp”, <http://www.electroimpulso.com.ar/ENERGIASOLAR/CONTROLADORES%20DE%20CARGA/REG12V10A.html>, última visita 12/12/2015.
- [63] SolaryEolica, “Procedimiento de instalación de paneles solares”, <http://www.solaryeolica.com.ar/contents/es/d168.html>, última visita 12/12/2015.
- [64] Antusol, “Instalación de sistema fotovoltaico”, <http://antusol.webcindario.com/instalacion.html>, última visita 12/12/2015.
- [65] Lorena Tobares, “La importancia y el futuro del biogás en la Argentina”, [http://www.petrotecnica.com.ar/1\\_2013/Petrotecnica/PdfsSinPublic/LaImportancia.pdf](http://www.petrotecnica.com.ar/1_2013/Petrotecnica/PdfsSinPublic/LaImportancia.pdf), última visita 12/12/2015.
- [66] Jorge Hilbert, “Manual para la producción de biogás, Instituto de Ingeniería Rural” I.N.T.A., [http://inta.gob.ar/sites/default/files/script-tmp-manual\\_para\\_la\\_produccion\\_de\\_biogs\\_del\\_iir.pdf](http://inta.gob.ar/sites/default/files/script-tmp-manual_para_la_produccion_de_biogs_del_iir.pdf), última visita 12/12/2015.
- [67] <http://www.fao.org/about/es/>, última visita 12/12/2015.
- [68] Ministerio De Planificación Federal, Inversión Pública Y Servicios, “Proyecto de Energías Renovables: Diagnóstico, Resultados y Conclusiones” [http://www.infoleg.gov.ar/basehome/actos\\_gobierno/actosdegobierno21-12-2009-1.htm](http://www.infoleg.gov.ar/basehome/actos_gobierno/actosdegobierno21-12-2009-1.htm), última visita 12/12/2015.
- [69] Universidad Nacional de Colombia sede Palmira, “Fundamentos para el Diseño de Biodigestores” <http://www.bdigital.unal.edu.co/7967/4/luisoctaviogonzalezsalcedo.20121.pdf>, última visita 12/12/2015.
- [70] <http://bio-digestores.blogspot.com.ar/2012/06/tipos-de-biodigestores.html>, última visita 12/12/2015.

- [71] <http://www.eg-ingenieria.com.ar/> , última visita 12/12/2015.
- [72] EG- Ingeniería, “Biodigestor de desplazamiento horizontal”, <http://www.eg-ingenieria.com.ar/biodigestores-desplazamiento-horizontal.html>, última visita 12/12/2015.
- [73] EG- Ingeniería, “Biodigestor con membranas”, <http://www.eg-ingenieria.com.ar/biodigestores-con-geomembranas.html>, última visita 12/12/2015.
- [74] EG- Ingeniería, “Biodigestor prefabricados”, <http://www.eg-ingenieria.com.ar/biodigestores-prefabricados.html>, última visita 12/12/2015.
- [75] Geo Soluciones, “Biodigestores”, <http://www.geomembranas.com.co/geoproductos/biodigestores>, última visita 12/12/2015.
- [76] Diego Avendaño Allen-Perkins, “Diseño y construcción de un digestor anaerobio de flujo de pistón que trate los residuos generados en una explotación ganadera de la localidad de Loja, Ecuador, empleando tecnologías apropiadas” [http://www.gessa-ex.es/documentos/publicaciones/guia\\_odt.pdf](http://www.gessa-ex.es/documentos/publicaciones/guia_odt.pdf), última visita 12/12/2015.
- [77] Carsten Bormann, “Getting Started with IPv6 in Low-Power Wireless “Personal Area” Networks (6LoWPAN)”, <https://www.iab.org/activities/workshops/smartobjects/tutorial/>, última visita 12/12/2015.
- [78] <http://www.atmel.com/tools/SAM3X-EK.aspx>, última visita 05/01/2016.
- [79] <http://www.atmel.com/devices/sam3x8e.aspx>, última visita 05/01/2016.
- [80] <http://www.telit.com/products/product-service-selector/product-service-selector/show/product/gl865-quad/>, última visita 05/01/2016.
- [81] Everlight, “5mm Phototransistor PT334-6C”, <http://www.everlight.com/file/ProductFile/PT334-6C.pdf>, última visita 01/05/2015.
- [82] <http://www.elster.com.ar/es/index.html>, última visita 01/05/2015.
- [83] <http://www.myeel.com.ar/>, última visita 01/05/2015.
- [84] [http://www.myeel.com.ar/producto.php?producto=Sonda\\_para\\_Lectura](http://www.myeel.com.ar/producto.php?producto=Sonda_para_Lectura)
- [85] Republica del Peru, Ministerio de Energias y Minas , “Plan Nacional de Electrificación Rural (PNER)”, [http://dger.minem.gob.pe/ArchivosDger/PNER\\_2015-2024/F1-PNER-2015-2024.pdf](http://dger.minem.gob.pe/ArchivosDger/PNER_2015-2024/F1-PNER-2015-2024.pdf), última visita 01/05/2015.
- [86] Subsecretaría de Desarrollo Regional y Administrativo, Ministerio del Interior y Seguridad Pública, Republica de Chile, “ Programa de electrificación rural – PER: Contrato de préstamos 1475/OC-CH”, <http://www.subdere.gov.cl/documentacion/programa-de-electrificaci%C3%B3n-rural-contrato-de-pr%C3%A9stamo-1475oc-ch>, última visita 01/05/2015.
- [87] Ministeria de Agricultura Ganadería y Pesca, Republica Argentina, “PROYECTO: “ELECTRIFICACIÓN RURAL PARA EL DESARROLLO PECUARIO DEL DEPARTAMENTO 25 DE MAYO” [http://www.prosap.gov.ar/docs/SJuan\\_Electrificacion25DeMayo-PAA.pdf](http://www.prosap.gov.ar/docs/SJuan_Electrificacion25DeMayo-PAA.pdf), última visita 01/05/2015.
- [88] Guillermina P. Jacinto, María Luciana Nogar, “Electrificación rural, desarrollo territorial y pequeñas localidades. El caso de Tandil (provincia de Buenos Aires, Argentina)” [http://www.filo.unt.edu.ar/rev/ieq/ieq\\_21/Breves%201\\_electrificacion\\_jacinto\\_nogar.pdf](http://www.filo.unt.edu.ar/rev/ieq/ieq_21/Breves%201_electrificacion_jacinto_nogar.pdf), última visita 01/05/2015.
- [89] B. van Campen, D. Guidi y G. Best, “Energía solar fotovoltaica para la agricultura y desarrollo rural sostenibles”, Documento de Trabajo sobre Medio Ambiente y Recursos Naturales, No.3 FAO, Roma, 2000, <http://www.fao.org/uploads/media/Solar%20photovoltaic%20for%20SARD%20ES.pdf>, última visita 01/05/2015.
- [90] Canziani, Mónica; Gomez, Rodrigo; Lupi, Daniel; Nassipián, Verónica; Slawiski Javier; Turconi, Diego; Zaradnik, Ignacio, “ Plataforma de conexión de Redes Eléctricas Inteligentes a Internet de las Cosas”, CASE 2014, Agosto 2014.
- [91] ST,” STM32F405xx STM32F407xx, ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera”, <http://www.st.com/st-web->

- [ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf](http://ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf), última visita 01/05/2015.
- [92] ATMEL, "Atmel M90E36A, Enhanced Poly-Phase High-Performance Wide-Span Energy Metering IC", <http://www.atmel.com/Images/Atmel-46004-SE-M90E36A-Datasheet.pdf>, última visita 01/05/2015.
- [93] Silabs, "Si8640/41/42/45, LOW-POWER QUAD-CHANNEL DIGITAL ISOLATOR", <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si864x.pdf>, última vista 01/05/2015.
- [94] Talema, "AC1020 - 20 Amp Current Transformer" <http://www.nuvotem.com/en/products/pdf/AC-1020%20Jun-06.pdf>, última visita 01/05/2015.
- [95] Unit-T, "UT208", <http://uni-trend.com/UT208.html>, última visita 01/05/2015.
- [96] Unit-T, "UT71A", <http://uni-trend.com/UT71A.html>, última visita 01/05/2015.
- [97] <http://latinamerica.cat.com/cda/files/2061850/9/LSXE0007-00.pdf>, última visita 01/05/2015.
- [98] Schneider Electric, "Transferencia de redes", <http://www.schneider-electric.cl/documents/local/catalogos/de/cap4.pdf>, última visita 01/05/2015.
- [99] Editores Online, "Sistema de transferencia automático - Siemens" [http://www.editores-srl.com.ar/revistas/ie/255/sistema\\_de\\_transferencia\\_automatgico](http://www.editores-srl.com.ar/revistas/ie/255/sistema_de_transferencia_automatgico), última visita 01/05/2015.
- [100] Electra Molins, "Cuadri automático AUT-MP", <http://www.electramolins.es/AUT-MP12.aspx?MenuSup=7>, última visita 01/05/2015.
- [101] Velasquez Ingenieros Asociados S.A.S, "Transferencia automática con contactores", [http://www.velasquez.com.co/paginas/transferencia\\_automatgica\\_con\\_contactores.php](http://www.velasquez.com.co/paginas/transferencia_automatgica_con_contactores.php), última visita 01/05/2015.
- [102] CPI S.A, "Sistema de transferencia automática", <http://cpi.com.ar/soluciones/sistema-de-transferencia-automatgica/>, última visita 01/05/2015.
- [103] Montatixe Almachi, Walter Patricio; Pillajo Guano, Aníbal Geovanny "Construcción de un tablero de transferencia automática de energía eléctrica para la central telefónica de Echandía de Andinatel S.A." <http://bibdigital.epn.edu.ec/handle/15000/2082>, última visita 01/05/2015.
- [104] Carabajal Gutierrez José Gualberto, "Control automático de transferencia de energía eléctrica" <http://es.slideshare.net/PedroChavez1/control-automtico-de-transferencia-de-energa-elctrica>, última vista 01/05/2015.
- [105] Hans Eslava Maldonado, Nicolas Franco Franco, "Automatización de una planta de emergencia para cargas no mayores a 10kw", <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/10105/1/6.pdf>, última visita 01/05/2015.
- [106] Secretaria de Energía del Ministerio de Economía y Finanzas, "Energía Eólica", [http://www.energia.gov.ar/contenidos/archivos/publicaciones/libro\\_energia\\_eolica.pdf](http://www.energia.gov.ar/contenidos/archivos/publicaciones/libro_energia_eolica.pdf), última visita 12/12/2015.
- [107] CAMMESA, <http://portalweb.cammesa.com/default.aspx>, última visita 12/12/2015.
- [108] Edenor S.A, "Consumo de Artefactos Eléctricos" [http://www.edenor.com.ar/cms/SP/CLI/HOG/USO\\_consumo.html](http://www.edenor.com.ar/cms/SP/CLI/HOG/USO_consumo.html), última visita 12/12/2015.
- [109] <http://www.clasf.mx/generador-electrico-eolico-700w-max12v-en-tijuana-5862545/>, última visita 12/12/2015.
- [110] <http://articulo.mercadolibre.com.mx/MLM-505587224-generador-eolico-600w-12v24v-JM>, última visita 12/12/2015.
- [111] <http://www.ebay.es/itm/700W-24V-i-700-WINDGENERATOR-von-iSTA-BREEZE-WINDKRAFTANLAGE-WIND-TURBINE-1->, última visita 12/12/2015.

- [112] Hugh Piggott, "Manual de auto construcción de generadores eólicos"
- [113] Johans Toledo, "Proyecto Aerogeneradores", <http://es.scribd.com/doc/285281582/Proyecto-Aerogeneradores#scribd>, última visita 12/12/2015.
- [114] Fondear S.R.L., "Elija su generador eólico", [http://www.fondear.org/infonautic/Equipo\\_y\\_Usos/Equipamiento/GeneradorEolico/GeneradorEolico.htm](http://www.fondear.org/infonautic/Equipo_y_Usos/Equipamiento/GeneradorEolico/GeneradorEolico.htm), última vista 12/12/2015.
- [115] Adaptación Fondear S.R.L., "Todo sobre Baterías", [http://www.fondear.org/infonautic/Equipo\\_y\\_Usos/Electricidad\\_Energia/Baterias/Baterias.htm](http://www.fondear.org/infonautic/Equipo_y_Usos/Electricidad_Energia/Baterias/Baterias.htm), última vista 12/12/2015.
- [116] Información obtenida por el material teórico brindado por la cátedra "Tecnología Electrónica", la cual forma parte del quinto año de la carrera Ingeniería Electrónica en la Universidad Nacional de la Matanza.
- [117] Información obtenida por el material teórico brindado por la cátedra "Tecnología Electrónica", la cual forma parte del quinto año de la carrera Ingeniería Electrónica en la Universidad Nacional de la Matanza.
- [118] <http://www.miscel.dk/MiscEl/miscel.html>, última vista 12/12/2015.
- [119] Texas Instruments, "xx555 Precision Timers", <http://www.ti.com/lit/ds/symlink/nc555.pdf>, última vista 12/12/2015.
- [120] [https://en.wikipedia.org/wiki/Paradox\\_\(database\)](https://en.wikipedia.org/wiki/Paradox_(database)), última vista 12/12/2015.
- [121] Arvind Pradham, "About Paradox Database", <http://www.dbtalks.com/UploadFile/e6aced/about-paradox-database/>, última vista 12/12/2015.
- [122] <https://es.wikipedia.org/wiki/MySQL>, última vista 12/12/2015.
- [123] Ibrahim Ersoy, "MySQL FAQs", <http://www.dbtalks.com/uploadfile/iersoy/mysql-faqs/> última vista 12/12/2015.
- [124] MySQL, "MySQL 5.7 Reference Manual / ... / Limits on Table Size" <http://dev.mysql.com/doc/refman/5.7/en/table-size-limit.html> , última vista 12/12/2015.
- [125] [https://en.wikipedia.org/wiki/Oracle\\_Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation), última vista 12/12/2015.
- [126] Oracle, "Información Confiable - Oracle Database 11g: Información General sobre las Nuevas Características" <http://www.oracle.com/technetwork/es/documentation/317497-esa.pdf>, última vista 12/12/2015.
- [127] Oracle, "ORACLE DATABASE 10G STANDARD EDITION" <http://www.oracle.com/technetwork/es/documentation/317499-esa.pdf>, última vista 12/12/2015.
- [128] Oracle, "Home / Database / Oracle Database Online Documentation, 10g Release 2 (10.2) / Grid Computing - Physical Database Limits" [https://docs.oracle.com/cd/B19306\\_01/server.102/b14237/limits002.htm#i287915](https://docs.oracle.com/cd/B19306_01/server.102/b14237/limits002.htm#i287915), última vista 12/12/2015.
- [129] <https://itxdesign.com/mysql-vs-oracle/>, última vista 12/12/2015.
- [130] <http://www.elconspirador.com/2013/12/17/stress-mysql-vs-oracle/>, última vista 12/12/2015.